

UNIVERSIDAD AUTONOMA DE MADRID

ESCUELA POLITECNICA SUPERIOR



Grado en Ingeniería Informática

TRABAJO FIN DE GRADO

**PORTAL WEB DEDICADO AL APRENDIZAJE
PROGRESIVO DE MECANOGRAFÍA**

Javier Senit Velasco
Tutor: Germán Montoro Manrique

JULIO 2020

PORTAL WEB DEDICADO AL APRENDIZAJE PROGRESIVO DE MECANOGRAFÍA

AUTOR: Javier Senit Velasco
TUTOR: Germán Montoro Manrique

Dpto. Ingeniería Informática
Escuela Politécnica Superior
Universidad Autónoma de Madrid
Julio 2020

Resumen (castellano)

Este Trabajo Fin de Grado se basa en la elaboración de un sitio web llamado QWERTY! cuya finalidad es el aprendizaje desde cero sobre mecanografía.

Este curso está dividido en dos partes claramente diferenciadas. La primera es un curso con el aprendizaje sobre todos los aspectos de la mecanografía: posición de las manos sobre el teclado, con qué dedo teclear cada letra, etc. Este aprendizaje se realizará a partir de diversos ejercicios con diferentes dificultades, que deberán pasarse con un mínimo de pulsaciones por segundo y un mínimo de errores para considerarse apto y poder pasar al siguiente ejercicio. La segunda se basa en un juego donde se tratará de escribir un texto largo en el menor tiempo posible y cometiendo el menor número de errores posibles.

En la primera parte, los ejercicios están separados por dificultades. Hay 3 dificultades: ejercicios básicos, ejercicios intermedios y ejercicios avanzados. Por lo general los ejercicios más básicos se basan en el aprendizaje de 2 o 3 letras, mientras que a medida que avanza la dificultad se van añadiendo más letras y, por consiguiente, se hace más difícil el llegar al mínimo de pulsaciones por segundo y no cometer errores.

La segunda parte se ha realizado de manera que cuando un usuario termine de escribir el texto largo aleatorio su resultado se guarde en una base de datos. Esto hace posible la visualización de un ranking por tiempos de todas las personas que hayan escrito dicho texto. Tanto el guardado de datos como la autenticación en la web funcionan a través de Firebase.

Abstract (English)

This Bachelor Thesis is based on the development of a web page called QWERTY! whose purpose is learning from scratch about typing.

This course is divided into two clearly differentiated parts. The first is a course with learning about all aspects of typing: hand position on the keyboard, with which finger to type each letter, etc. This learning will be carried out from various exercises with different difficulties, which must be passed with a minimum of beats per second and a minimum of errors to be considered suitable and to be able to proceed to the next exercise. The second is based on a game where you will try to write a long text in the shortest time possible and making the fewest mistakes possible.

In the first part, the exercises are separated by difficulties. There are 3 difficulties: basic exercises, intermediate exercises and advanced exercises. In general, the most basic exercises are based on learning 2 or 3 letters, while as the difficulty progresses more letters are added and, therefore, it becomes more difficult to reach the minimum number of beats per second. to make mistakes.

The second part has been done so that when a user finishes typing the long random text, their result is saved in a database. This makes it possible to view a time-based ranking of all the people who have written this text. Both data saving and web authentication works through Firebase.

Palabras clave (castellano)

Sitio web, Javascript, Firebase, HTML, CSS, JSON.

Keywords (inglés)

Web page, Javascript, Firebase, HTML, CSS, JSON.

Agradecimientos

A mi tutor Germán Montoro, por toda la ayuda que me ha brindado con el TFG y por aceptar mi propuesta del mismo.

A los profesores que he tenido durante toda mi estancia en la universidad, por todos los conocimientos que me han enseñado a lo largo de mi carrera.

A mis amigos, tanto de dentro como de fuera de la universidad, por todo el apoyo que me han dado durante la carrera y los buenos momentos que hemos compartido durante mi formación.

Y de una manera más personal, agradecer a mi familia por todo el apoyo, paciencia y ayuda que me han brindado en todo momento.

A todos vosotros, gracias.

INDICE DE CONTENIDOS

1	Introducción.....	1
1.1	Motivación	1
1.2	Objetivos	3
1.3	Organización de la memoria	4
2	Estado del arte.....	5
2.1	Cursos online.....	5
2.2	Javascript nativo frente a frameworks orientados	5
2.3	Comparativa con otras páginas webs	6
3	Diseño.....	8
3.1	Alcance de la aplicación	8
3.2	Análisis de requisitos.....	8
3.3	Subsistemas.....	8
3.3.1	Subsistema de registro	9
3.3.2	Subsistema de autenticación.....	9
3.3.3	Subsistema de consulta de perfil.....	9
3.3.4	Subsistema de contraseñas	9
3.3.5	Subsistema de cambio de contraseña	9
3.3.6	Subsistema de guardado de resultados.....	9
3.3.7	Subsistema de visualización de resultados.....	9
3.3.8	Subsistema de accesibilidad	9
3.3.9	Subsistema de actualización y modificación de resultados.....	10
3.4	Requisitos funcionales	10
3.4.1	Listado de requisitos funcionales según el subsistema	10
3.4.1.1	Subsistema de registro	10
3.4.1.2	Subsistema de autenticación	10
3.4.1.3	Subsistema de consulta de perfil	11
3.4.1.4	Subsistema de contraseñas.....	11
3.4.1.5	Subsistema de cambio de contraseña	11
3.4.1.6	Subsistema de guardado de resultados	11
3.4.1.7	Subsistema de accesibilidad.....	11
3.4.1.8	Subsistema de actualización y modificación de resultados	12
3.5	Requisitos no funcionales	12
3.6	Resolución de requisitos	13
3.6.1	Resolución de requisitos funcionales.....	13
3.6.1.1	Subsistema de registro	13
3.6.1.2	Subsistema de autenticación	13
3.6.1.3	Subsistema de consulta perfil.....	13
3.6.1.4	Subsistema de contraseñas.....	13
3.6.1.5	Subsistema de cambio de contraseña	13
3.6.1.6	Subsistema de guardado de resultados	13
3.6.1.7	Subsistema de accesibilidad.....	13
3.6.1.8	Subsistema de actualización y modificación de resultados	14
3.6.2	Resolución de requisitos no funcionales	14
3.7	Firebase como base de datos.....	14
4	Desarrollo	15
4.1	Estructura del proyecto	15
4.2	Aspecto visual general de la web	15

4.2.1 La importancia del color en una web	16
4.2.1.1 Color blanco en la web	17
4.2.1.2 Color azul en la web	17
4.2.1.3 Color naranja en la web	17
4.2.1.4 Color negro en la web	17
4.2.2 Tipografía empleada	17
4.3 Estructura de las escenas	17
4.4 Conjunto de escenas de la web	18
4.4.1 Autenticación/Registro	18
4.4.1.1 Scripts importantes usados	19
4.4.2 Selección de modo	21
4.4.3 Curso online: Selección de temario	22
4.4.4 Curso online: Selección de ejercicio	23
4.4.5 Curso online: Ejercicio	24
4.4.5.1 Scripts importantes usados	25
4.4.6 Juego online	26
4.4.6.1 Scripts importantes usados	27
4.4.7 Juego online: Ranking	29
4.4.7.1 Scripts importantes utilizados	30
4.4.8 Juego online: estadísticas	31
4.4.8.1 Scripts importantes utilizados	31
4.4.9 ¿Qué es QUERTY!?	32
4.5 Scripts comunes a todas las escenas	32
4.5.1 Incremento y decremento del tamaño de letra	32
4.5.2 Modo oscuro	33
4.5.3 Etiqueta <i>logged as</i> y cerrar sesión	33
5 Integración, pruebas y resultados	35
5.1 Pruebas de caja negra	35
5.2 Pruebas de caja blanca	35
5.3 Pruebas de validación	35
5.4 Validación de código limpio	35
6 Conclusiones y trabajo futuro	37
6.1 Conclusiones	37
6.2 Trabajo futuro	37
Referencias	39
Glosario	40
Anexos	I
Manual de usuario	I

INDICE DE FIGURAS

FIGURA 1-1. SESIONES DIARIAS EN LAS DIFERENTES INDUSTRIAS EN 2018	2
FIGURA 1-2. PORCENTAJE DE REBOTE EN LAS WEB DE LOS DIFERENTES SECTORES	3
FIGURA 4-1. ESTRUCTURA DEL PROYECTO.....	15
FIGURA 4-2. PSICOLOGÍA DE LOS COLORES	16
FIGURA 4-3. ESCENA DE AUTENTICACIÓN	18
FIGURA 4-4. ESCENA DE REGISTRO DE USUARIO.....	19
FIGURA 4-5. FUNCIÓN DE CREACIÓN DE USUARIO EN LA BASE DE DATOS DE FIREBASE.....	20
FIGURA 4-6. USUARIOS REGISTRADOS EN LA WEB VISTOS DESDE FIREBASE	20
FIGURA 4-7. FUNCIÓN QUE AUTENTICA A UN USUARIO EN LA WEB.....	21
FIGURA 4-8. FUNCIÓN DE CAMBIO DE CONTRASEÑA	21
FIGURA 4-9. ESCENA DE SELECCIÓN DE MODO	22
FIGURA 4-10. ESCENA DE SELECCIÓN DE TEMARIO	22
FIGURA 4-11. ESCENA DE SELECCIÓN DE EJERCICIO	23
FIGURA 4-12. ESCENA DE REALIZACIÓN DE EJERCICIO	24
FIGURA 4-13. RESULTADOS OBTENIDOS EN EJERCICIO	25
FIGURA 4-14. CÁLCULO DEL CÓDIGO ESPECÍFICO DE LA “Ñ” Y LA TILDE	26
FIGURA 4-15. ESCENA DE JUEGO ONLINE	26
FIGURA 4-16. FUNCIÓN QUE TRANSFORMA DATOS A FORMATO JSON.....	28
FIGURA 4-17. FUNCIÓN QUE SUBE AL SERVIDOR FIREBASE DATOS EN FORMATO JSON.....	28
FIGURA 4-18. ENTRADA DE LA BASE DE DATOS DE FIREBASE.....	28
FIGURA 4-19. RANKING DE UN TEXTO CON LOS USUARIOS REGISTRADOS EN ÉL	29
FIGURA 4-20. FUNCIÓN QUE OBTIENE DATOS DE LA REALTIME DATABASE DE FIREBASE	30
FIGURA 4-21. ESTADÍSTICAS GENERALES DE UN USUARIO	31
FIGURA 4-22. FUNCIÓN QUE OBTIENE EL ÍNDICE DE CADA LIBRO ESCRITO POR UN USUARIO.....	31
FIGURA 4-23. ESCENA QUE EXPLICA LA FINALIDAD DE LA WEB.....	32

<i>FIGURA 4-24. EJEMPLO DE MODO OSCURO EN LA ESCENA DE SELECCIÓN DE TEMARIO.....</i>	<i>33</i>
<i>FIGURA 4-25. FUNCIÓN QUE OBTIENE EL USUARIO ACTIVO Y LO ESCRIBE EN LA PARTE SUPERIOR DERECHA</i>	<i>34</i>
<i>FIGURA 4-26. FUNCIÓN QUE CIERRA LA SESIÓN DE UN USUARIO</i>	<i>34</i>
<i>FIGURA 5-1. RESULTADOS TRAS PASAR LA HERRAMIENTA JSHINT AL CÓDIGO DE REGISTRO</i>	<i>36</i>
<i>FIGURA A-1. ESCENA DE AUTENTICACIÓN Y REGISTRO DEL USUARIO.....</i>	<i>I</i>
<i>FIGURA A-2. ESCENA DE SELECCIÓN DE MODO.....</i>	<i>II</i>
<i>FIGURA A-3. ESCENA DE SELECCIÓN DE TEMARIO</i>	<i>II</i>
<i>FIGURA A-4. ESCENA DE SELECCIÓN DE EJERCICIO.....</i>	<i>III</i>
<i>FIGURA A-5. ESCENA DE REALIZACIÓN DE EJERCICIO.....</i>	<i>III</i>
<i>FIGURA A-6. EJEMPLO DE EJERCICIO NO SUPERADO.</i>	<i>IV</i>
<i>FIGURA A-7. ESCENA DE TEXTO LARGO DEL MODO DE JUEGO ONLINE.</i>	<i>IV</i>
<i>FIGURA A-8. ESCENA DE RANKING ONLINE DE UN TEXTO DETERMINADO.</i>	<i>V</i>
<i>FIGURA A-9. ESCENA DE ESTADÍSTICAS GENERALES DE UN USUARIO.</i>	<i>V</i>

1 Introducción

En este apartado del documento se realiza una descripción general del proyecto, explicando en primer lugar la motivación por la cual se ha realizado este proyecto, los objetivos que se han intentado conseguir y los que se han conseguido.

1.1 Motivación

Esta memoria de TFG busca poder explicar y describir todo el proceso que se ha llevado a cabo para la realización del sitio web. Tratará de explicar cómo se ha desarrollado (con que tecnologías y lenguajes) y el funcionamiento de ésta.

La motivación principal para la realización de este lenguaje es poder enseñar a todo el mundo a poder escribir en el ordenador de manera eficiente y sin cometer muchos errores. Se busca este objetivo ya que en muchas ocasiones el tiempo que se emplea para escribir un texto y después repasarlo para comprobar que no contiene errores es muy alto. Si desde un principio se empieza por tener una buena colocación sobre el teclado, saber teclear cada letra con el dedo que corresponda, etc, en el futuro la habilidad para poder escribir textos largos será mucho mejor y más eficaz.

Es por esto comentado anteriormente, que el sitio web está destinada principalmente a estudiantes de primaria y secundaria, donde cada vez más se está empezando a usar el ordenador como base para la enseñanza. Esto no impide que cualquier persona, tenga la edad que tenga, pueda usar esta web para formarse y aprender sobre mecanografía.

En la actualidad, las páginas web orientadas a la educación están cada vez cogiendo más fuerza. En un principio, internet fue creada principalmente como una herramienta ligada principalmente a la educación, donde poder instruirse y aprender de manera mucho más fácil y rápida.

Muchos de los analistas más antiguos relacionados al mundo de internet afirmaban en un principio que las características principales de internet concordaban con los intereses centrales de la educación. Por ejemplo, tanto la educación como internet, buscan la creación de contenido, el intercambio de información y la comunicación.

Pero con el tiempo se ha comprobado que, aunque en gran parte, internet sí se ha convertido en un lugar educativo donde aprender, han surgido muchos otros sectores donde internet tiene más influencia, como por ejemplo en el mundo de los deportes, noticias de la actualidad, etc.

En la **Figura 1-1** obtenida de digishuffle [\[1\]](#) se muestra las visitas diarias a diferentes webs según el tipo de industria.

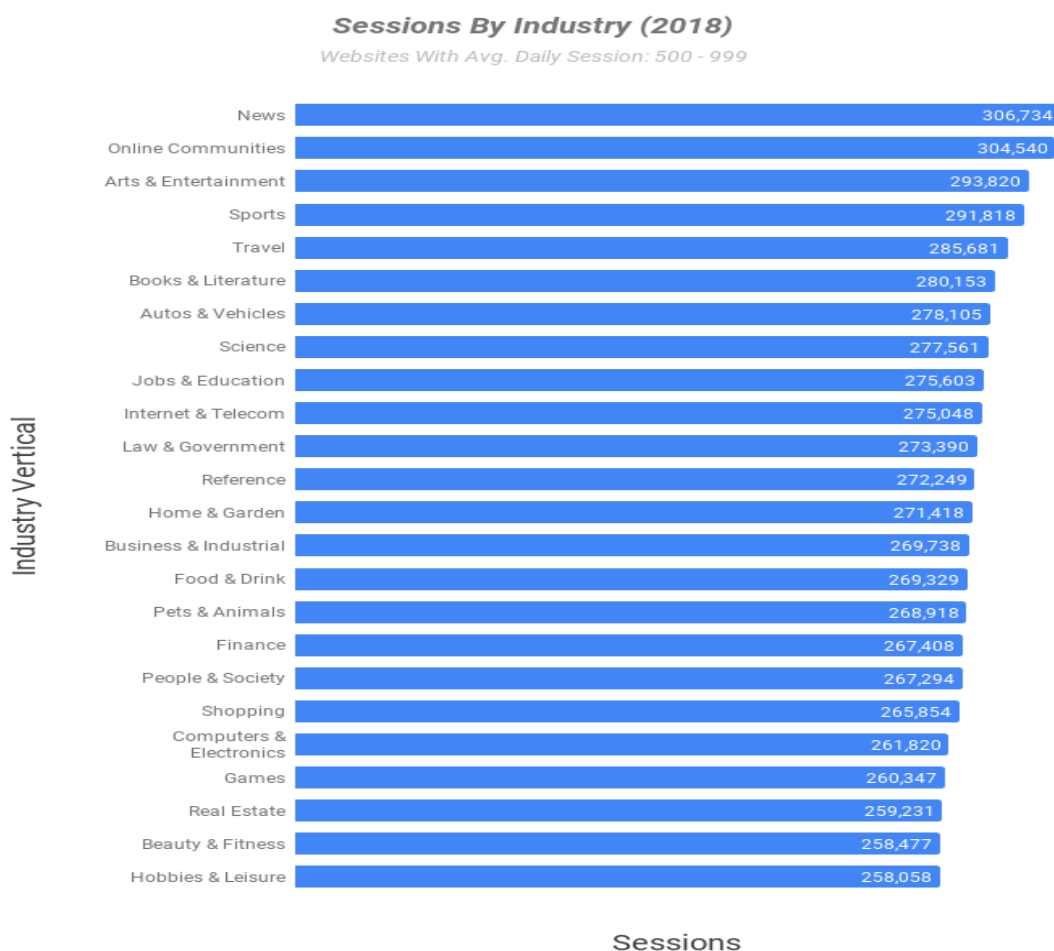


Figura 1-1. Sesiones diarias en las diferentes industrias en 2018

Las entradas que se realizan sobre páginas web relacionadas sobre enseñanza y educación son bastante altas si se engloban en ellas las industrias de ciencia, educación y trabajo; aunque no están tan altas como otras industrias como el deporte o los viajes.

Es por esto que se ha decidido realizar este proyecto, para intentar impulsar aún más la enseñanza y educación a través de internet, y comenzar a instalar estos métodos en colegios e institutos para una mejor formación.

Cabe destacar que la industria de la enseñanza en internet es una de las que menos *bounce rate* tiene en relación a las demás. *Bounce rate* es un término utilizado en los análisis del tráfico de visitantes de las web de Internet. Un rebote (en inglés *bounce*) se produce cuando un navegante abandona el sitio después de haber visto una sola página web, en unos pocos segundos. En la **Figura 1-2** obtenida de digishuffle [\[1\]](#) se muestran el *bounce rate* de cada industria, donde se puede observar que la educación, aunque no sea la que menor ratio tiene, tampoco es de las más altas.

Bounce Rate vs Industry Vertical

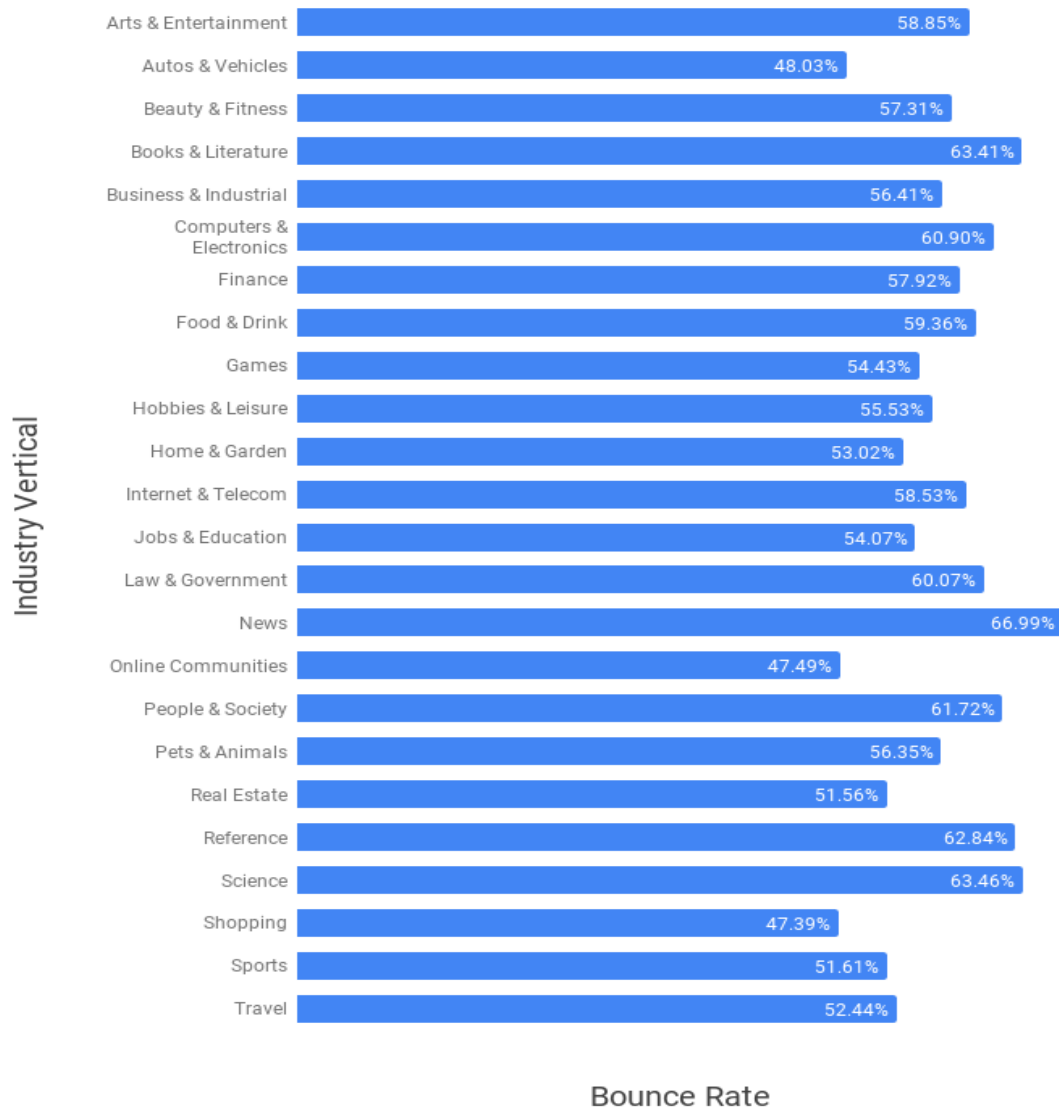


Figura 1-2. Porcentaje de rebote en las web de los diferentes sectores

1.2 Objetivos

Los objetivos principales de este proyecto son los siguientes:

- Elaboración de un sitio web orientada que apoye a la educación y aprendizaje para que los más jóvenes tenga en un futuro mejores habilidades y puedan trabajar de manera mucho más eficiente.
- Creación de ejercicios con dificultad creciente: ejercicios básicos, intermedios y avanzados.
- Diseño sencillo y enfocado directamente en lo que se quiere enseñar.

- Crear una competición sana entre usuarios que busquen obtener el mejor resultado en el modo de juego del portal.

1.3 Organización de la memoria

La memoria consta de los siguientes capítulos:

- Estado del arte: En este capítulo se realiza una compilación de resultados de sitios web relacionados con la de este proyecto y se describe sus similitudes y diferencias, así como una explicación del porqué de las tecnologías escogidas para la realización del proyecto.
- Diseño: En este capítulo se detalla cada una de las funcionalidades que se diseñaron para este sitio web, así como el diseño de toda la información que se guarda en Firebase y de qué forma se hace.
- Desarrollo: En este capítulo se explica cómo se ha ido realizando cada una de las funcionalidades del proyecto de manera interna, enfocándose en que estrategias y algoritmos se han ido usando en cada momento.
- Integración, pruebas y resultados: En este capítulo se detallaran aspectos relacionados sobre que navegadores funciona de manera correcta el sitio web, así como las pruebas que se realizaron para verificar su correcto funcionamiento.
- Conclusiones y trabajo futuro: En este capítulo se dan unas conclusiones finales sobre la elaboración del proyecto y se explican modificaciones y mejoras futuras que se podrían realizar sobre el mismo.

2 Estado del arte

Existen millones de páginas web en la actualidad desarrolladas en el ámbito de la educación y la enseñanza. Las hay de diferentes tipos como por ejemplo las divulgativas, que se basan en explicar diversos temas de haciendo uso de texto y videos. También existen las plataformas LMS (*Learning Managment System*) como por ejemplo Moodle[2], que es un sistema de gestión de cursos de distribución libre que tiene como finalidad ayudar a los educadores y profesores a crear comunidades de aprendizaje en línea. Entre otros tipos también se encuentran los cursos online, los cuales se centran en un tema y lo explican a partir de ejercicios y pruebas. Es en este tipo de páginas web donde se engloba este proyecto.

2.1 Cursos online

Los cursos online se basan principalmente en una versión moderna de los cursos tradicionales que se imparte en escuelas, institutos y universidades de manera presencial.

En estos cursos se puede compartir contenido de aprendizaje de una forma ordenada y organizada que permite a los usuarios ir aprendiendo y progresando de manera incremental sobre un determinado tema.

Se basan básicamente en un plan de estudios organizado en diferentes temas y unidades, terminado con algún proyecto final donde se debe demostrar todo lo aprendido durante el curso.

Este proyecto se basa en crear un sitio web que es un curso online sobre mecanografía. Este curso todo los pasos anteriormente comentados, posee un temario con diversos ejercicios, y una prueba final, la cual se corresponde en este caso con un juego online con otros usuarios, al cual solo se accede cuando se tiene cierto nivel sobre mecanografía.

En la actualidad España es uno de los países de la Unión Europea donde más repercusión y seguimiento tienen los cursos online[3]. La formación online es un aspecto donde muchos países de la Unión Europea no destacan demasiado y es por eso que se le daba dar más visibilidad para ser un método de enseñanza que combine con la tradicional enseñanza presencial.

2.2 Javascript nativo frente a frameworks orientados

Javascript es el lenguaje funcional más popular del mundo. JavaScript es y siempre ha sido, al menos desde la versión 1.2, un lenguaje de programación funcional[4]. Una aplicación desarrollada en javascript nativo quiere decir que la funcionalidad está escrita directamente en archivos javascript que están unidos a un html para darle funcionalidad. Todo esto se realiza sin ningún tipo de entorno de programación que ayude al programador durante el desarrollo de la aplicación o sitio web. Algunas de las webs más importantes hechas en javascript nativo son

Por otro lado, un framework es eso, una herramienta que puede aprovechar un desarrollador para crear aplicaciones y páginas web avanzadas, especialmente SPA(*Single Page Application*). Las aplicaciones SPA, tal como se explica en el artículo de desarrolloweb [5], son aquellas donde todo el código HTML, CSS y JS se carga una única vez, dando la ventaja de que se puede navegar por los diferentes apartados de la página mucho más rápido, pues el contenido está precargado anteriormente por completo. Para realizar estas SPA suele hacerse uso de diferentes frameworks como por ejemplo: Angular, React, Ember.js, etc. De igual manera los frameworks pueden usarse para el desarrollo de todo tipo de aplicaciones, no solo SPA, aunque sí que están bastante orientados a este tipo.

El sitio web QWERTY! está realizado con javascript nativo. Las características por las que se decidió emplear javascript nativo frente a frameworks fueron las siguientes:

- Javascript seguirá existiendo aunque el framework desaparezca o no se le proporcionen actualizaciones. En un futuro, lo más probable es que javascript siga existiendo y sacando nuevas versiones y actualizaciones. Esto no es seguro con los frameworks, pues pueden ser abandonados por diferentes razones y con ello dejen de usarse. Un programa escrito en nativo desarrollado a día de hoy, funcionará casi seguro en un navegador web dentro de 10 años. Por supuesto uno escrito a partir de un framework también puede seguir funcionando, pero hay más posibilidades de que esto no ocurra.
- El código realizado con un framework es entendible solo para aquellos que han estudiado ese framework, mientras que una aplicación en javascript nativo es entendida tanto por los que saben y los que no saben sobre frameworks. Es decir, javascript native es más conocido mundialmente.
- Muchos problemas que surgen con los frameworks son debido al poco control que se tiene sobre el código y es por esto que en muchas ocasiones no se pueden solucionar de manera rápida y eficaz.

2.3 Comparativa con otras páginas webs

Existen diversas páginas que ofrecen cursos de mecanografía como el que se ha realizado para este proyecto. Los hay tanto gratis como de pago. Los más completos son los de pago, pero el problema básico de éstos es básicamente ese, hay que pagar para realizarlo. Esto no es nada malo, pues la gran mayoría de cursos que se imparten en el mundo no son gratis.

Este proyecto es un curso online gratuito al que cualquier persona podrá acceder simplemente registrándose en la página. Por supuesto también existen muchos cursos de mecanografía gratuitos, como por ejemplo ARTypist[6] o TypingClub[7]. La principal diferencia que existe entre este proyecto y estas otras páginas de mecanografía es que éstas no guardan un progreso ni los resultados que obtiene un usuario para poder verlos en un futuro.

Otra característica propia de este proyecto es que aparte de ser un curso de aprendizaje, tiene como parte final de éste un juego online con otros jugadores que hará que cada usuario quiera mejorar cada vez más para lograr ser el mejor y el más rápido.

Existen páginas dedicadas a esto como por ejemplo KeyHero[\[8\]](#) o Velocidactil[\[9\]](#), pero ninguna que contenga el conjunto de curso y juego online a la vez. Esto hace que QWERTY! Sea una web diferente e innovadora.

3 Diseño

En este capítulo se explicarán todos los detalles que se han seguido durante el desarrollo del proyecto, relacionados con el diseño de éste y cuáles son sus requisitos funcionales y no funcionales.

En primer lugar se detallará el alcance de la aplicación, después se mostrará el análisis de requisitos, y por último la solución propuesta y cuáles de los requisitos se han cumplido correctamente.

3.1 Alcance de la aplicación

El público al que va destinado este proyecto es todo a aquel que entienda y comprenda el lenguaje castellano, ya que el curso está diseñado para aprender a mecanografiar en castellano, pues la mecanografía en otros idiomas cambia drásticamente ya que por ejemplo, en Inglés se usa con frecuencia el apóstrofo y en francés los acentos graves y circunflejos, caracteres que en castellano no se usan nunca.

En cuanto a que tecnologías se necesitarán para acceder a este curso, solo será necesario un ordenador con conexión a internet. El navegador que se emplee será indiferente, pues el proyecto se ha realizado de tal forma que funcione perfectamente en todos los navegadores más famosos y conocidos, como lo son Chrome, Firefox, Safari, Explorer u Ópera.

El proyecto va dirigido mayoritariamente a personas jóvenes, pues en parte, se quiere que este curso sea parte de la educación durante los años de colegio o instituto. Esto no quita que el curso pueda realizarse por personas de cualquier edad, pues todo el mundo puede querer aprender cuando quiera.

3.2 Análisis de requisitos

El análisis de requisitos es una etapa muy importante durante el desarrollo de un proyecto de este tipo, pues unos requisitos definidos de manera errónea desde un principio, puede acarrear problemas muy serios en el futuro.

Los requisitos se han separado según el subsistema al que pertenecen y según si son funcionales o no funcionales.

3.3 Subsistemas

Cada uno de los subsistemas que se definen a continuación tiene el objetivo de englobar todas las funcionalidades del proyecto en diferentes secciones, para poder diferenciarlas de manera correcta y así trabajarlas mejor en un futuro

3.3.1 Subsistema de registro

El subsistema de registro engloba toda aquella funcionalidad relacionada con el registro de nuevos usuarios en el sitio web. Esto incluye el guardado de su información en la base de datos, que en este caso es Firebase.

3.3.2 Subsistema de autenticación

El subsistema de autenticación contiene toda la funcionalidad relacionada con el inicio de sesión de cada usuario en la web y la obtención de su información personal de la base de datos la cual será usada para identificarse mientras navega por la web.

3.3.3 Subsistema de consulta de perfil

El subsistema de consulta de perfil agrupa todas aquellas funciones que podrá realizar un usuario para ver todos sus datos personales.

3.3.4 Subsistema de contraseñas

El subsistema de contraseñas engloba las funciones que se encargan de definir como debe ser la contraseña que introduzca el usuario, los símbolos que se permiten y su longitud.

3.3.5 Subsistema de cambio de contraseña

El subsistema de cambio de contraseña comprende las funciones relacionadas con la posibilidad que se le da al usuario de cambiar de contraseña introduciendo su correo electrónico actual.

3.3.6 Subsistema de guardado de resultados

El subsistema de guardado de resultados contiene toda la funcionalidad relacionada al guardado de los tiempos y pulsaciones por segundo que realiza cada usuario en los diferentes textos.

3.3.7 Subsistema de visualización de resultados

El subsistema de visualización de resultados engloba toda la funcionalidad necesaria para que un usuario autenticado pueda ver sus resultados, tanto los de los ejercicios realizados durante el curso, como su posición en el ranking en cada texto.

3.3.8 Subsistema de accesibilidad

El subsistema de accesibilidad contiene todas las funciones relacionadas con las ayudas que se proporcionan para usuarios con alguna discapacidad o que simplemente quieren cambiar el diseño de la web. Estas son por ejemplo el modo oscuro, el modo daltónico, o la posibilidad de cambiar el tamaño de la letra en cualquier momento.

3.3.9 Subsistema de actualización y modificación de resultados

El subsistema de actualización y modificación de datos engloba aquellas funciones que se encargan de actualizar los resultados de un usuario si este resuelve el ejercicio con un tiempo mejor y con menos errores.

3.4 Requisitos funcionales

Los requisitos funcionales son las descripciones explícitas de cómo debe comportarse una solución de software y que información debe manejar [\[10\]](#).

Los requisitos funcionales deben satisfacer todas aquellas funciones que se desea que el proyecto cumpla y deben proporcionar una descripción sencilla y fácil de entender para que posteriormente su solución se lleve a cabo mejor y de manera más eficiente.

Para que los requisitos sean correctos no deben tener ambigüedades, pues esto podría provocar interpretaciones erróneas que conlleven muchos problemas en el futuro.

3.4.1 Listado de requisitos funcionales según el subsistema

3.4.1.1 Subsistema de registro

RF-1: El sistema deberá permitir el registro de cualquier usuario siempre y cuando este rellene de manera correcta el formulario correspondiente.

RF-2: El sistema comprobará si el usuario ha introducido un correo electrónico válido a la hora de rellenar el formulario.

RF-3: El sistema deberá preguntar una segunda vez por la contraseña para comprobar que es exactamente la que el usuario desea.

RF-4: El sistema comprobará que el correo electrónico que introduce el usuario para registrarse no está ya en la base de datos. Si esto ocurre no se creará la cuenta del usuario y se solicitará una nueva.

3.4.1.2 Subsistema de autenticación

RF-5: El sistema deberá dejar al usuario iniciar sesión introduciendo el correo y la contraseña que usó para registrarse.

RF-6: El sistema comprobará que el correo introducido para autenticarse existe en la base de datos. Si no existe se le dará una nueva oportunidad al usuario para que introduzca un correo correcto.

RF-7: El sistema comprobará si la contraseña introducida se corresponde con el correo electrónico mirándolo en la base de datos. Si no se corresponde con la contraseña correcta, se le permite introducir de nuevo ambos campos, correo y contraseña.

3.4.1.3 Subsistema de consulta de perfil

RF-8: El sistema deberá dejar al usuario visualizar la información de su perfil.

3.4.1.4 Subsistema de contraseñas

RF-9: El sistema deberá solicitar una contraseña de una longitud mínima de 6 caracteres para evitar posibles robos.

RF-10: La contraseña no deberá ser visible con caracteres legibles durante su introducción. Deberán usarse símbolos que reemplacen la contraseña original, como por ejemplo asteriscos o puntos.

RF-11: El sistema guardará toda la información personal de los usuarios en Firebase, de manera que solo un administrador pueda acceder a ellas.

3.4.1.5 Subsistema de cambio de contraseña

RF-12: El sistema permitirá a un usuario cambiar su contraseña siempre y cuando proporcione un correo electrónico válido que esté guardado en la base de datos.

3.4.1.6 Subsistema de guardado de resultados

RF-13: El sistema deberá guardar los resultados de los ejercicios del curso online que realice cada usuario para que éstos puedan visualizarlos después.

RF-14: El sistema deberá guardar los resultados que cada usuario han realizado sobre un texto en el modo juego para posteriormente poder mostrar un ranking con los mejores tiempos.

RF-15: El sistema deberá decirle a un usuario si sus resultados son aptos o no para que éste decida si seguir con ejercicios más avanzados o no.

3.4.1.7 Subsistema de accesibilidad

RF-16: El sistema deberá permitir cambiar entre modo normal y modo oscuro en cualquier momento y en cualquier pantalla.

RF-17: El sistema deberá permitir cambiar entre el modo normal y el modo daltónico en cualquier momento y en cualquier pantalla.

RF-18: El sistema deberá permitir aumentar y disminuir el tamaño de la letra en cualquier momento y en cualquier pantalla.

3.4.1.8 Subsistema de actualización y modificación de resultados

RF-19: El sistema deberá poder actualizar los resultados de cada usuario si estos consiguen mejores resultados en cuanto a tiempo y número de errores.

3.5 Requisitos no funcionales

Los requisitos no funcionales son requisitos que no se refieren directamente a las funciones específicas suministradas por el sistema (características de usuario), sino a las propiedades del sistema: rendimiento, seguridad, disponibilidad [\[11\]](#).

RNF-1: El sistema no deberá dejar en ningún momento ver las contraseñas de otros usuarios desde ningún punto de la web.

RNF-2: El sistema generará un token de sesión para cada usuario con el fin de mantener más seguros los datos privados de cada usuario.

RNF-3: El sistema deberá proporcionar una interfaz que debe sencilla e intuitiva, fácil de usar.

RNF-4: El sistema deberá estar lo suficientemente seguro como para no recibir ciberataques, ya sean por ejemplo mediante SQL injection o mediante el acceso remoto a la base de datos de Firebase.

RNF-5: El sistema no tardará más de 2 segundo en realizar la carga inicial de cada una de las páginas.

RNF-6: El sistema deberá tener un tiempo de respuesta con el servidor Firebase rápido.

RNF-7: El sistema deberá adaptar el contenido del sitio web si el usuario decide minimizar o maximizar el tamaño de la ventana del navegador donde la ha abierto.

RNF-8: El sistema deberá permitir una correcta visualización del contenido como mínimo en los siguientes navegadores: Chrome, Firefox, Safari, Explorer y Ópera.

RNF-9: El sistema deberá poder abrir el sitio web en móviles o tabletas, aunque la finalidad de la web sea totalmente opuesta al uso en estos dispositivos.

RNF-10: La sesión de un usuario se mantendrá activa durante al menos 2 horas después de cerrar el navegador sin salir de su sesión.

RNF-11: El sitio web no se bloqueará en ningún momento durante la petición de datos al servidor.

3.6 Resolución de requisitos

Durante el desarrollo de este proyecto se han cumplido satisfactoriamente la mayor parte de los requisitos.

3.6.1 Resolución de requisitos funcionales

En esta sección se comentarán que requisitos funcionales se han cumplido y cuáles no, y cuál ha sido el motivo

3.6.1.1 Subsistema de registro

Se han cumplido todos los requisitos.

3.6.1.2 Subsistema de autenticación

Se han cumplido todos los requisitos.

3.6.1.3 Subsistema de consulta perfil

Se han cumplido todos los requisitos

3.6.1.4 Subsistema de contraseñas

Se han cumplido todos los requisitos

3.6.1.5 Subsistema de cambio de contraseña

Se han cumplido todos los requisitos

3.6.1.6 Subsistema de guardado de resultados

Se han cumplido todos los requisitos excepto el RF-13. No se ha implementado esta funcionalidad pues se decidió que mostrándole información al usuario tras terminar cada ejercicio no era necesario guardar dicha información.

3.6.1.7 Subsistema de accesibilidad

Se han cumplido todos los requisitos excepto el RF-17. Debido al poco conocimiento sobre el daltonismo, se decidió no implementar esta funcionalidad. Podría haberse implementado

con colores un poco al azar y que no funcionase de nada para aquellos que padecen la enfermedad.

3.6.1.8 Subsistema de actualización y modificación de resultados

Se han cumplido todos los requisitos atendiendo a la actualización de datos en la parte del juego online. Se decidió que los datos sobre los ejercicios del curso no se guardaran.

3.6.2 Resolución de requisitos no funcionales

Los requisitos no funcionales que no se han cumplido han sido los siguientes:

RNF-4: Tarea bastante difícil de realizar de manera completamente perfecta.

RNF-9: No se ha comprobado la compatibilidad con todo tipo de móviles y tabletas ni con sus diferentes tamaños.

3.7 Firebase como base de datos

Durante el desarrollo de este proyecto se ha usado Firebase para guardar toda la información de los usuarios.

Tal como se explica en rockcontent, Firebase es una plataforma digital que se utiliza para facilitar el desarrollo de aplicaciones móviles o web de una forma efectiva, rápida y sencilla, la cual es utilizada por sus diversas funciones como una técnica de marketing digital para incrementar la base de usuarios y generar más beneficios económicos[\[12\]](#).

Una de las principales ventajas que nos ofrece Firebase es su colección de funciones de *Firebase Authentication*, las cuales permiten de manera muy sencilla realizar el registro y la autenticación de los usuarios.

Los datos relacionados con resultados del juego online se han guardado en la Realtime Database de Firebase, la cual deja guardar datos en formato JSON. Estos datos se explicarán en el capítulo Desarrollo de este documento.

4 Desarrollo

En este capítulo se explicará todo el desarrollo que se ha ido realizando a lo largo del proyecto, así como todas las diferentes funciones implementadas y el aspecto final de cada archivo html con sus archivos javascripts y css asociados.

4.1 Estructura del proyecto

En la **Figura 4-1** se muestra la organización en carpetas que se ha realizado para el desarrollo del proyecto.

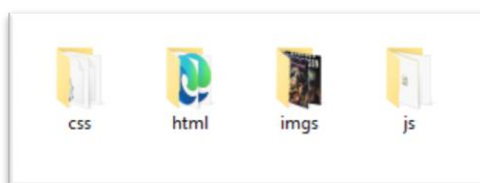


Figura 4-1. Estructura del proyecto

- **Css:** En esta carpeta se encuentran todas las hojas de estilos que se han usado para realizar la web. Estas son las encargadas de dar un mejor aspecto visual al sitio web.
- **Html:** En esta carpeta se encuentran todos los ficheros html realizados. En estos ficheros se encuentra la parte estática visible (solo texto) de cada una de las páginas que componen el proyecto.
- **Imgs:** En esta carpeta se encuentran todas las imágenes usadas en el proyecto.
- **Js:** En esta carpeta se concentra la mayor parte del trabajo del proyecto. Todas las funciones se han realizado en ficheros js, los cuales aportan funcionalidad a los ficheros html.

4.2 Aspecto visual general de la web

La web está creada de tal forma que cada una de las páginas que la contienen tenga un aspecto visual parecido y que lo haga distintivo de otras páginas. Cabe destacar que el resultado no es del todo profesional, pues es el proyecto ha sido realizado por un estudiante y no por un diseñador web.

Uno de los apartados más importantes a la hora de diseñar una web, es el cómo se verá visualmente. Si el aspecto es poco profesional, es muy probable que el usuario acabe abandonando la web antes de tiempo. En mi caso lo más complicado de escoger fueron los colores y la tipografía de la letra.

4.2.1 La importancia del color en una web

El color en una web es muy importante. Cada color genera una reacción diferente en cada persona. A esto se le llama psicología de los colores. La **Figura 4-2** obtenida de kapitalsoluciones[13] muestra que valores transmiten a las diferentes personas cada color:

BLANCO	Pureza, inocencia, optimismo, frescura, limpieza, simplicidad
ROJO	Fortaleza, pasión, determinación, deseo, amor, fuerza, valor, impulsividad
NARANJA	Calidez, entusiasmo, creatividad, éxito, ánimo
AMARILLO	Energía, felicidad, diversión, espontaneidad, alegría, innovación
VERDE	Naturaleza, esperanza, equilibrio, crecimiento, estabilidad, celos
AZUL	Libertad, verdad, armonía, fidelidad, progreso, seriedad, lealtad
PÚRPURA	Serenidad, místico, romántico, elegante, sensual, ecléctico
ROSA	Dulzura, delicadeza, exquisited, sentimientos de gratitud, amistad
GRIS	Paz, tenacidad
NEGRO	Silencio, sobriedad, poder, formalidad, misterio

Figura 4-2. Psicología de los colores

Tras pensar mucho tiempo que colores se escogerían para el proyecto, se eligieron principalmente 4: blanco, azul, naranja y negro.

Con el blanco se intenta mostrar que el trabajo realizado es simple pero con un objetivo claro.

Con el azul se hace referencia al tema del que trata la web, el cual es tecnológico y del mundo de la informática.

Con el naranja se intenta hacer ver que es un proyecto realizado por una persona joven y con mucha energía.

Con el negro se intenta transmitir que el proyecto es serio y a la vez elegante.

4.2.1.1 Color blanco en la web

El color blanco es el color de fondo de todas y cada una de las escenas. Esto proporciona mucha limpieza visual y hace que el usuario pueda centrarse mejor a la hora de realizar los ejercicios y de jugar online.

4.2.1.2 Color azul en la web

El color azul es el color del *header* de la web. Este es un color que se verá en todas y cada una de las diferentes escenas y que será el característico de la web.

Además de ser el color del *header* será el que se muestre cuando se pulse el modo oscuro en cualquiera de las escenas. Se ha escogido este color porque minimiza bastante el daño a la vista cuando el usuario se encuentra en un espacio sin luz.

4.2.1.3 Color naranja en la web

Es el color que se ha escogido para casi todas las letras que contiene la web. Es un color vistoso y fácil del leer, no como pudieran ser un amarillo o un verde claro, que en muchas ocasiones hace imposible la lectura del texto.

4.2.1.4 Color negro en la web

El color negro es el que se ha escogido para el fondo de las diferentes figuras y ventanas importantes que tiene la web.

4.2.2 Tipografía empleada

La tipografía empleada para la web es *AbeeZee* en muchos tamaños diferentes.

4.3 Estructura de las escenas

La organización que se ha escogido para la realización de la web es la siguiente:

- Header: En el header aparecerá el logo de la web: QWERTY!
- Sidebar izquierdo: En el panel izquierdo se mostrarán las opciones de accesibilidad y el acceso a las estadísticas de cada usuario.

- Content: Parte central de cada escena. Aquí se muestra la mayoría del contenido de la web.
- Footer: En el footer se muestra el creador de la web, abajo a la derecha.

4.4 Conjunto de escenas de la web

En este capítulo se mostrarán todas las escenas del proyecto a las que se pueden acceder simplemente navegando a través de ellas.

Se explicará en profundidad que tiene cada una de ellas, sus funciones más importantes y los problemas que surgieron a la hora de su desarrollo.

4.4.1 Autenticación/Registro

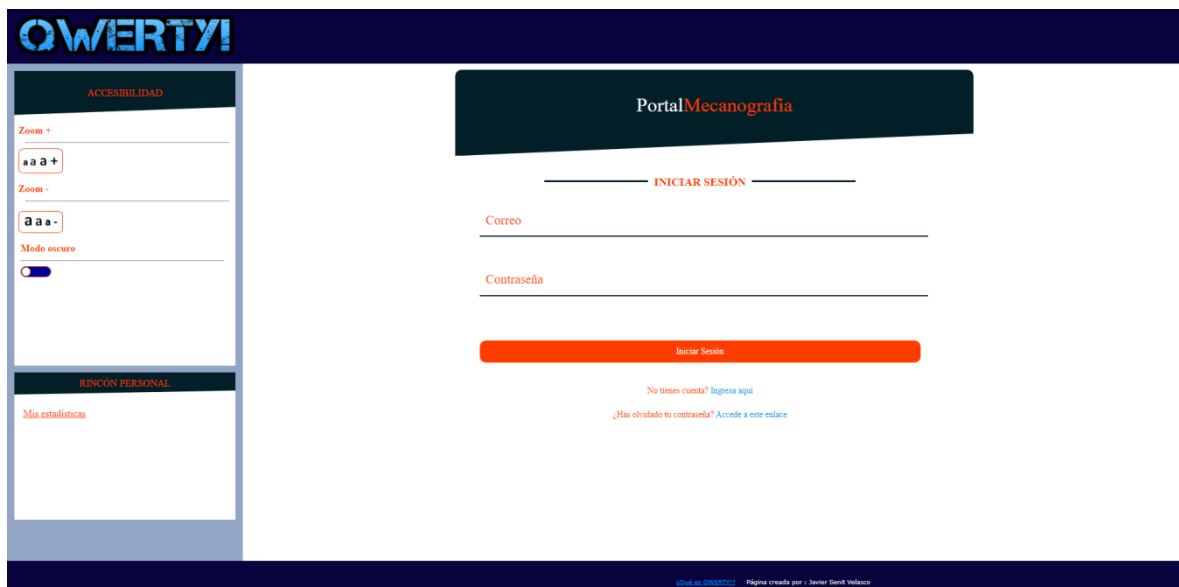


Figura 4-3. Escena de autenticación

La **Figura 4-3** es la primera escena que nos encontramos tras acceder a *QWERTY!*. Se trata de una escena de autenticación donde el usuario deberá introducir su correo electrónico y su contraseña para acceder al curso. Si el usuario no posee ninguna cuenta aún solo tiene que hacer click en el enlace que hay debajo del botón de iniciar sesión.

Si el usuario pulsa dicho botón el formulario cambiará de aspecto, pero la url seguirá siendo la misma. El nuevo se muestra en la **Figura 4-4**.

El formulario ahora ha cambiado, solicitando al usuario su correo, y repita 2 veces una contraseña para poder registrarse.

Si todos los campos son introducidos de manera correcta se guardará una entrada en la base de datos de Firebase de dicho usuario.

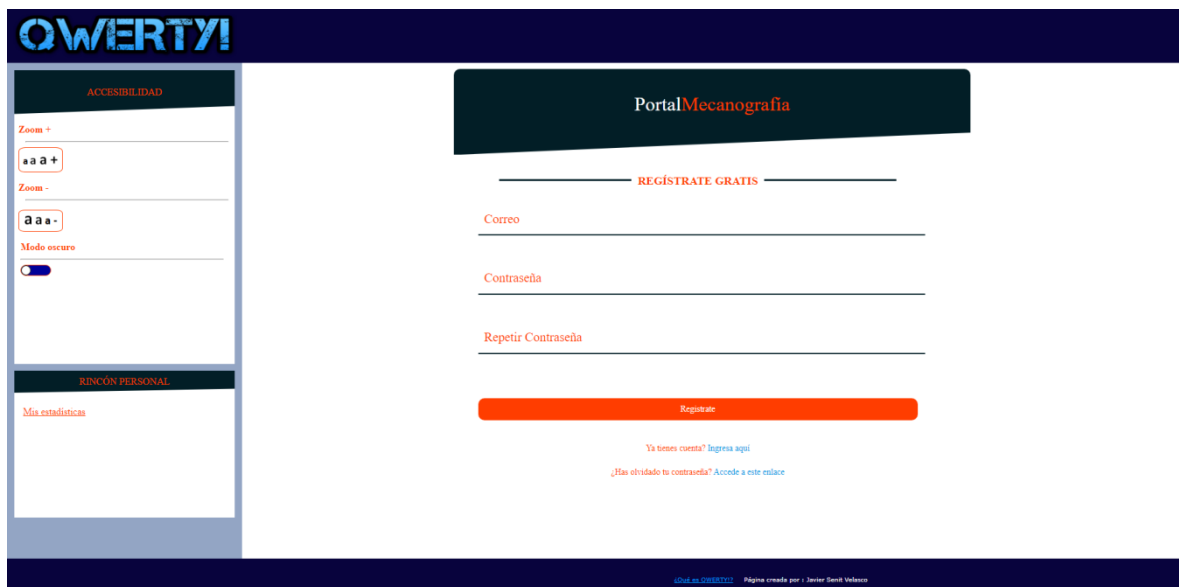


Figura 4-4. Escena de registro de usuario

Cuando el usuario se registre se cargará de nuevo el formulario anterior para que se pueda autenticar y poder entrar al curso.

Como se observa a la izquierda, existen dos menús. El primero de ellos es el de accesibilidad que se encontrará en todas y cada una de las escenas. Contiene 3 elementos clickables, uno de ellos para incrementar el tamaño de la letra, otro para disminuirlo y otro para cambiar al modo oscuro. El segundo menú permite ver las estadísticas del usuario en los rankings online del modo juego de la web. Si el usuario no está autenticado no podrá acceder a esta funcionalidad y se le dará un aviso para que se autentique.

4.4.1.1 Scripts importantes usados

En esta subsección se explican algunos de los scripts más importantes de esta escena.

4.4.1.1.1 Registro de usuario

Para realizar tanto el registro como la autenticación de los usuarios se han usado las funciones de la librería Firebase Authentication que nos proporciona Firebase.

El código que realiza el registro se muestra en la **Figura 4-5**. En primer lugar se comprueban las dos contraseñas introducidas en los campos de *Contraseña* y *Repetir Contraseña*. Si ambos son iguales se procede a usar la función `createUserWithEmailAndPassword(correo, pass)` la cual nos proporciona Firebase y sirve para crear una entrada de un usuario en la base de datos.

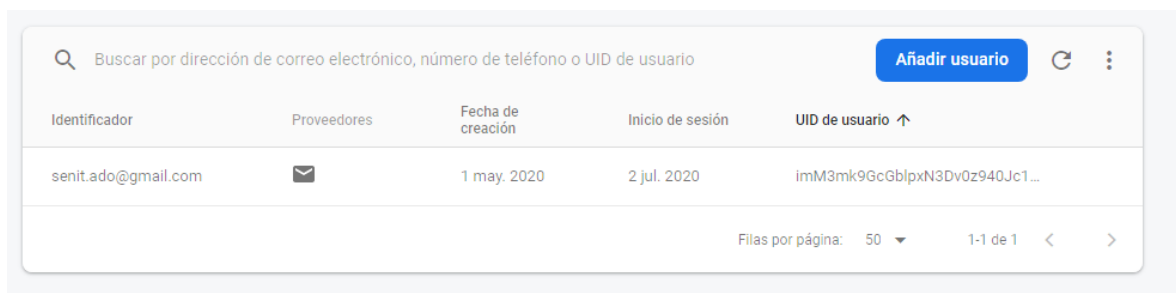
```


if(pass == passpass){
    firebase.auth().createUserWithEmailAndPassword(correo, pass).then(function(value) {
        console.log(value);
        var user = firebase.auth().currentUser;
        alert("Usuario creado correctamente: " + user.email);
    }).catch(function(error) {
        alert(error);
    });
}else{
    alert("Las contraseñas deben ser iguales");
}
}

```

Figura 4-5. Función de creación de usuario en la base de datos de Firebase

La entrada del usuario se guarda en la Firebase tal como la **Figura 4-6** lo muestra



Identificador	Proveedores	Fecha de creación	Inicio de sesión	UID de usuario ↑
senit.ado@gmail.com		1 may. 2020	2 jul. 2020	imM3mk9GcGblpxN3Dv0z940Jc1...

Filas por página: 50 1-1 de 1

Figura 4-6. Usuarios registrados en la web vistos desde Firebase

Como se puede observar, se guarda un identificador de usuario que es su email, el tipo de identificador que es de tipo email, la fecha de creación, el último inicio de sesión y la UID del usuario, la cuál será empleada para obtener un token que mantenga iniciada la sesión a través de todas las escenas de la web.

Estos datos no serán visibles nunca por parte de un usuario. Solo un administrador podrá acceder a ellos a través de Firebase.

4.4.1.1.2 Autenticación de usuario

La autenticación del usuario también se realiza con las funciones que nos proporciona la librería Firebase Authentication. En la **Figura 4-7** se puede ver el código que lo realiza.

```

        firebase.auth().signInWithEmailAndPassword(correo, pass)
        .then(function(firebaseUser) {
            var user = firebase.auth().currentUser;
            alert( "Logeado correctamente: " + user.email);
            window.location.href = "seleccion.html";
        })
        .catch(function(error) {
            alert(error);
        });
    }
}

```

Figura 4-7. *Función que autentica a un usuario en la web*

La función `signInWithEmailAndPassword(correo, pass)` comprueba si el usuario con el correo y la contraseña introducida se encuentra en la base de datos, y si es así le redirige a la escena principal de selección de modo.

4.4.1.1.3 Cambio de contraseña

Si se pulsa el enlace para cambiar la contraseña se abrirá una ventana auxiliar donde el usuario deberá introducir su correo electrónico. En la Figura 4.8 se muestra la función que realiza el envío de un mensaje de solicitud de cambio de contraseña. Este mensaje redirigirá al usuario a un página propia de Firebase donde introducir su nueva contraseña.

```

function cambiopassword(){

    firebase.auth.sendPasswordResetEmail().then(function() {
        // Email sent.
    }).catch(function(error) {
        // An error happened.
    });

}

```

Figura 4-8. *Función de cambio de contraseña*

La función `sendPasswordResetEmail()` es la que se encarga de realizar todo el proceso anteriormente comentado. Cabe destacar que si a esta se le introduce un email, no hará falta que el usuario lo haga. Esto sería útil si en enlace se encontrase en algún sitio una vez accedido al curso, donde el programa tendría acceso al email del usuario.

4.4.2 Selección de modo



Figura 4-9. Escena de selección de modo

En la escena de la Figura 4-9 se le da al usuario la opción de escoger entre los dos modos que contiene la web. El primero de ellos es el curso online, donde el usuario realizará ejercicios para ir mejorando su mecanografía poco a poco hasta llegar a escribir sin mirar el teclado. El segundo de ellos es el modo de juego online. Este botón redirigirá al usuario a un texto aleatorio que tendrá que escribir para posteriormente ver el ranking y su posición.

Como se observa arriba a la derecha, aparece el usuario que está actualmente autenticado en la web. Si desea cerrar sesión solo tendrá que pulsar el link *Cerrar Sesión*, que automáticamente redirigirá al usuario a la escena de Registro.

4.4.3 Curso online: Selección de temario

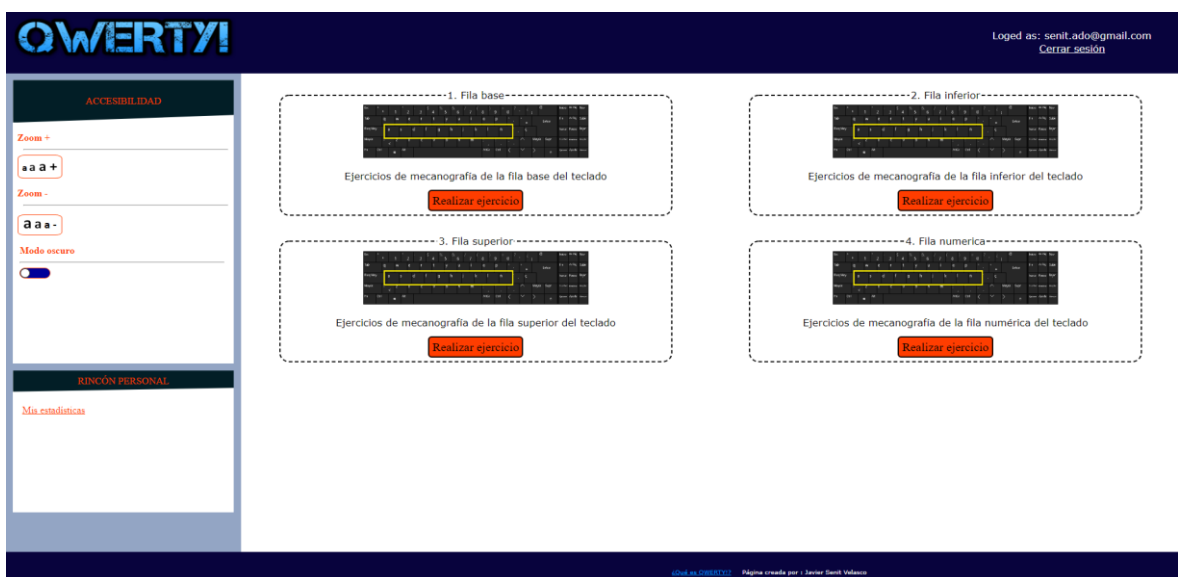


Figura 4-10. Escena de selección de temario

En la escena correspondiente a la **Figura 4-10** se le da al usuario la posibilidad de escoger entre 4 temarios diferentes. Como se observa existen 4 temas:

- Fila base: En este tema se realizarán ejercicios con la fila central de letras del teclado.
- Fila inferior: En este tema se realizarán ejercicios con la fila inferior de letras del teclado.
- Fila superior: En este tema se realizarán ejercicios con la fila superior de letras del teclado.
- Fila numérica: En este tema se realizarán ejercicios con la fila numérica del teclado.

4.4.4 Curso online: Selección de ejercicio



Figura 4-11. Escena de selección de ejercicio

La escena mostrada en la **Figura 4-11** es bastante simple. Según el temario al que se acceda aparecerán las tres opciones que se observan. Habrá ejercicios de nivel básico, de nivel intermedio y de nivel avanzado.

Cuando el usuario pase el ratón por uno de ellos se desplegarán todos los ejercicios disponibles para esa dificultad. Se ha decidido que el número de ejercicios en cada sección sea el siguiente:

- Ejercicios básicos: 5 ejercicios básicos donde se irán usando poca cantidad de letras o números a la vez (máximo 2).
- Ejercicios intermedios: 3 ejercicios intermedios donde el número de letras y números que se emplearán a la vez serán 4 o 5.

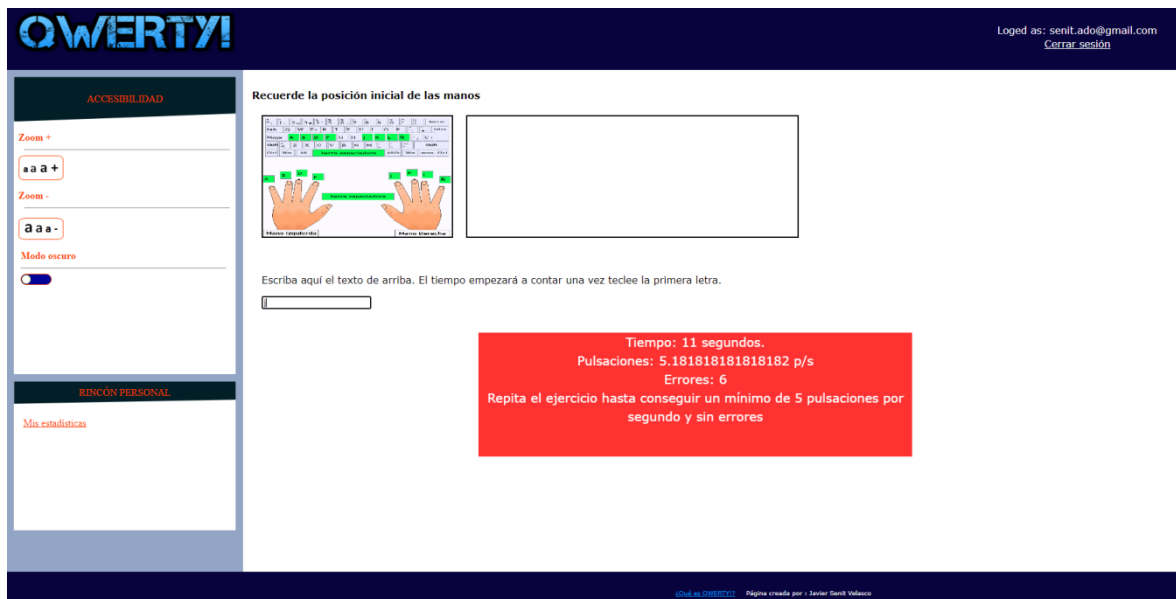


Figura 4-13. Resultados obtenidos en ejercicio

4.4.5.1 Scripts importantes usados

En esta subsección se muestran los scripts más importantes realizados para esta escena

4.4.5.1.1 Introducción del texto

Esta es la función principal de la escena. Debe ser capaz de reconocer todas las teclas que pulse el usuario sin fallos y sin retardo, ya que si hubiese retardo no se podría obtener un resultado en cuanto a tiempo correcto, y muchas veces la mayoría de las letras presionadas no se contarían.

En un comienzo se pensó en realizar esta acción con un *keydown* que captase que letra del teclado se había pulsado. Esta opción se rechazó por no cumplir con todos los requisitos que se querían ya que se comprobó que en muchas ocasiones, el usuario teclea más de una letra antes de levantar el dedo de la primera. La función *keydown* cuando ocurre esto no captaba la segunda tecla que se pulsaba, ya que espera siempre a que se realice un *keyup* de la primera para contar la siguiente pulsación.

Debido a esto, se pensó en realizarlo con la opción contraria, con un *keyup*, que contase cada una de las veces que una tecla era levantada tras su presión. Esta opción también fue rechazada. Si un usuario teclea como se ha dicho antes más de una tecla antes de levantar la primera, esta función no captaba la pulsación de la segunda hasta que la primera fuera levantada por completo.

Al final se tuvo que usar la función *keypress*, ya que era la única que no debía esperarse a que una tecla fuera levantada para empezar a contar la siguiente. El único problema que surgió con esta función fue que no reconocía la introducción de algunos caracteres

correctamente, como por ejemplo la “ñ” o la tilde “ ’ ”. Cuando introduces estos caracteres, la función entendía otros caracteres totalmente distintos debido a que no está configurada para reconocer todos caracteres españoles. Es por esto que se tuvo que realizar la operación matemática mostrada en la **Figura 4-14** que consiguiera el valor correspondiente del teclado español.

```
key = e.key;
letra = letra.slice(0, idx) + key + letra.slice(idx + Math.abs(0));
letra = letra.slice(-1);
```

Figura 4-14. Cálculo del código específico de la “ñ” y la tilde

En *key* se guardaba el valor real obtenido por la función y con la operación matemática se conseguía transformarlo para que se correspondiese con la “ñ” o con la tilde.

4.4.5.1.2 Animación de carga de resultado

Cuando el usuario termina de escribir el texto, aparece una animación de carga, simulando que se están calculando sus resultados. Esto es solo visual, pues los errores se van calculando a medida que va escribiendo, y el tiempo que ha tardado en realizar el ejercicio, se calcula poniendo una marca una vez se mete la primera letra, y otra marca cuando el texto a escribir queda vacío

4.4.6 Juego online

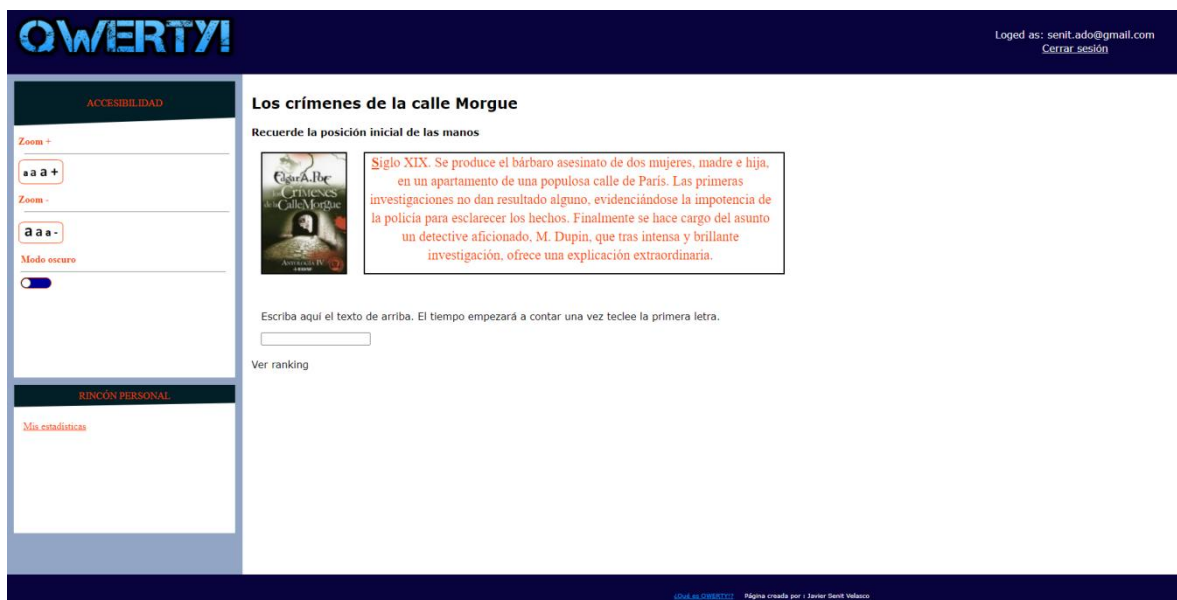


Figura 4-15. Escena de juego online

En la **Figura 4-15** se muestra la escena del juego online. Los textos que se han escogidos son resúmenes de libros al azar, cogidos de una base de datos propia. El hecho de coger resúmenes de libros como textos largos a escribir se decidió simplemente porque parecía

una buena práctica dar a conocer libros a los usuarios, de manera que pudieran interesarse por alguno de ellos a medida que iban escribiendo su resumen.

La mecánica es la misma que con los ejercicios, escribir el texto lo más rápido posible. La diferencia es que en este caso, una vez finalizada la escritura del texto, los resultados se guardarían en la base de datos de Firebase con el fin de realizar el ranking de usuarios. El usuario podrá pulsar sobre el link *Ver ranking* para acceder al ranking de usuarios que han escrito ese mismo texto.

Los datos que se guardarán en la base de datos serán el nombre del usuario, el tiempo que ha tardado en realizarlo y las pulsaciones por segundo que ha tenido.

4.4.6.1 Scripts importantes usados

En esta subsección se muestran los scripts más importantes realizados para esta escena

4.4.6.1.1 Introducción del texto

Dado que el diseño del juego online es el mismo que el de cualquier ejercicio del curso, se ha usado la misma función para el cálculo de las teclas pulsadas.

4.4.6.1.2 Guardado de datos del usuario

Los datos que se guardan del usuario, como ya se ha comentado anteriormente son, el nombre, el correo, el tiempo conseguido, y las pulsaciones por segundo realizadas. Estos datos deben guardarse en Firebase en modo de JSON.

4.4.6.1.2.1 Realtime Database

La Realtime Database es una funcionalidad que ofrece Firebase, la cual nos permite gestionar una base de datos dentro de sus servidores.

La base de datos que nos permite gestionar es no relacional (NoSQL). Tal como se explica en las guías de Firebase, todos los datos de Firebase Realtime Database se almacenan como objetos JSON. La base de datos puede definirse como un árbol JSON situado en la nube. A diferencia de una base de datos de SQL, no existen tablas ni registros. Cuando le agregas datos al árbol JSON, estos se convierten en un nodo de con una clave asociada[14].

Los archivos JSON no son más que archivos que guardan información con una sintaxis muy simple. Se basan en la estructuración de datos clave-valor. Es muy común que exista como valor de una clave, otro archivo JSON. Gracias a la función de la **Figura 4-16** se conseguía pasar datos a formato JSON

```
function arrayJSON(correo, time, pulsaciones){
    var data = {
        correo: correo,
        time:time,
        pulsaciones: pulsaciones
    };
    return data
}
}
```

Figura 4-16. Función que transforma datos a formato JSON

La funcionalidad es simple, se le pasan los parámetros que se quieren guardar en la base de datos, y te devuelve una variable con el formato JSON clave-valor especificado. Este formato es similar al de los diccionarios, muy usados por ejemplo en Java.

4.4.6.1.3 Subida de datos a Realtime Database

Una vez se han pasado los valores al formato JSON que es el que la Realtime Database de Firebase entiende, con la función de la **Figura 4-17** se suben los datos a la base de datos.

```
var arrayData= arrayJSON(correo, time, pulsaciones);
var task = firebase.database().ref(aleatorio + "/usuarios/" + correo);
task.set(arrayData);
```

Figura 4-17. Función que sube al servidor Firebase datos en formato JSON

En primer lugar, los datos se pasan al formato JSON con la función anteriormente explicada y se guardan en la variable arrayData. Tras esto hay que guardarse la referencia de la Realtime Database donde se van a introducir los datos. En este caso como se observa la referencia es `aleatorio + "/usuarios/" + correo`. Este aleatorio se refiere al identificador del libro que se ha escogido y del cual se ha escrito su resumen. Esto es necesario guardarlo ya que cada libro tiene su propio ranking individual. La **Figura 4-18** muestra el aspecto de la Realtime Database de Firebase cuando un usuario a terminado con éxito el texto número 16.



Figura 4-18. Entrada de la base de datos de Firebase

En primer lugar y como raíz, se muestra el nombre de la Realtime Database. Sus hijos directos son los identificadores de cada uno de los libros de los cuales los usuarios escriben

los textos en el modo online. Después se muestran todos los usuarios que han realizado cada texto, con su correo, nombre, pulsaciones totales y el tiempo conseguido.

Los libros de los cuales se han escogido los textos a escribir, también tiene su propio formato JSON. Poseen 4 claves con sus correspondientes valores, las cuales son, el id, el título del libro, el resumen de éste y una imagen asociada.

El hecho de que tanto el correo como el nombre esté escrito con punto y no con coma. Se debe a que la RealTime Database no acepta el punto como carácter. Es por esto que en ella se guardan los datos con puntos y después en el código simplemente se transforman por comas.

Esto se consigue con una función muy sencilla pero muy usada durante todo el proyecto, pues el obtener datos de la RealTime Database es muy normal como veremos en escenas posteriores.

Para subir datos se intercambian los puntos por comas: `correo = correo.split('.').join(',')`, mientras que para obtener datos se hace al revés: `correo = correo.split(',').join('.')`.

4.4.7 Juego online: Ranking



Figura 4-19. Ranking de un texto con los usuarios registrados en él

A la escena de la **Figura 4-19** se llega tras pulsar el enlace correspondiente desde la escena donde se realiza el juego online (donde se escribe el texto largo).

Se muestra una tabla con todos los usuarios que han resuelto completamente el texto indicado en la parte superior de la tabla. El ranking muestra las posiciones teniendo en cuenta el tiempo empleado. Se decidió esto porque el tener en cuenta las pulsaciones por segundo no servía de mucho, pues normalmente aquel que terminase antes el texto, por lo general, había conseguido más pulsaciones por segundo.

Se muestra en negrita aquel usuario que está autenticado actualmente, haciendo así más fácil la tarea al usuario de buscarse por toda la tabla, cuando los usuarios registrados en el texto fueran muchos.

En una primera versión se mostraban los correos electrónicos de los usuarios, pero se decidió que esto podría estar en contra de la privacidad de los mismos, por lo que se tuvo que mostrar solamente el nombre del usuario en la lista.

4.4.7.1 Scripts importantes utilizados

En esta subsección se muestran los scripts más importantes realizados para esta escena.

4.4.7.1.1 Obtención de datos de la Realtime Database

Este fue uno de los apartados más complejos de realizar. Esto se debe a que la función que nos aporta Firebase para la obtención de datos de la Realtime Database es asíncrona.

Según la página oficial de Firebase para recuperar los datos almacenados en Firebase Realtime Database, se debe adjuntar un objeto de escucha asíncrono a la referencia de una base de datos. El agente de escucha es activado una vez para el estado inicial de los datos y posteriormente otra vez cuando los datos cambian [15]. Esto provoca serios problemas a la hora de mostrar los datos de la base de datos en la web, ya que mientras Firebase realiza consultas a la Realtime Database, el código javascript sigue ejecutándose, finalizando a veces antes de que se obtuviesen todos los resultados. La función que realiza la obtención de resultados es la siguiente es la mostrada en la **Figura 4-20**.

```
firebase.database().ref(paramstr + "/usuarios").on("child_added", function(snapshot){  
    matriz.push(snapshot.val());  
});
```

Figura 4-20. Función que obtiene datos de la Realtime Database de Firebase

La función trabaja como un bucle for, va pasando hijo por hijo desde la referencia indicada y va guardando sus valores en la variable *snapshot*. Estos se van guardando en una matriz para después imprimirlos en una web. El problema ya comentado anteriormente hace que, al ser asíncrona la función, cuando realiza la primera iteración, el código siga ejecutándose hacia abajo, por lo que si se quiere imprimir algo, solo se imprime un valor, pues solo se ha realizado una iteración del falso bucle for.

Para solucionar esto, es en el interior de este bucle for donde se decidió escribir todo el código para la impresión en la web, de tal manera que cada vez que comenzaba una nueva iteración se borraba todo lo escrito en la web, para que, una vez llegada la ultima iteración, con todos los valores en la matriz, se recorriese esta y se escribieran correctamente todos los valores.

4.4.8 Juego online: estadísticas

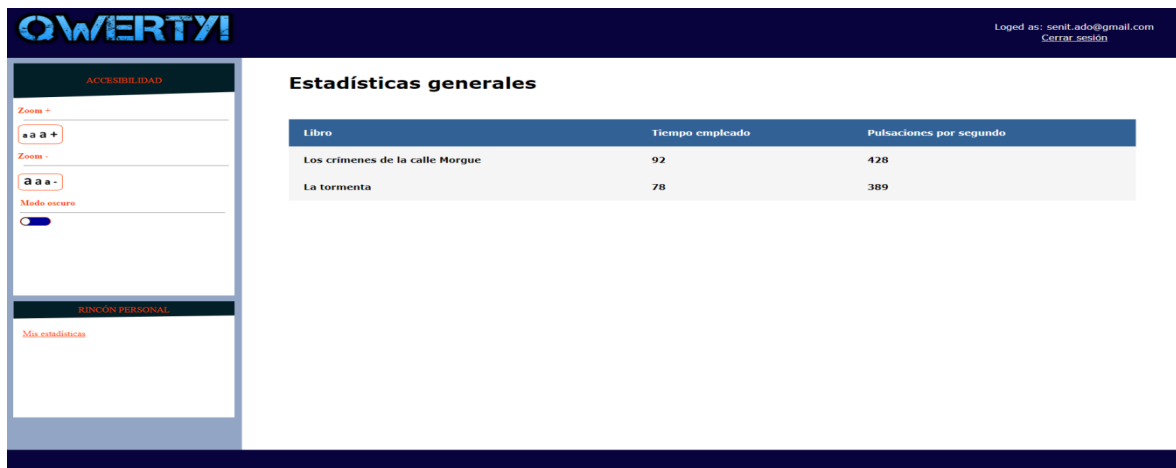


Figura 4-21. Estadísticas generales de un usuario

A la escena de la **Figura 4-21** se llega pulsando en enlace del apartado *Rincón Personal* del menú izquierdo. La función de esta escena es mostrarle al usuario todos los textos que ha completado y sus resultados.

Además podrá pulsar sobre el texto que quiera para acceder al ranking de dicho texto, mostrándose por tanto la escena de la **Figura 4-19** anteriormente comentada.

4.4.8.1 Scripts importantes utilizados

En esta subsección se muestran los scripts más importantes realizados para esta escena

4.4.8.1.1 Obtención de datos de la Realtime Database

En esta ocasión la función es muy parecida a la de la **Figura 4-20**.

```
firebase.database().ref().on("child_added", function(snapshot) {
    numero = snapshot.ref.key
});
```

Figura 4-22. Función que obtiene el índice de cada libro escrito por un usuario

La diferencia es que ya no se accede a la referencia ya no es la etiqueta *Usuarios*, sino que es la etiqueta raíz. Como se observa en la **Figura 4-22**, ésta referencia tiene como hijos directos los id de los libros. Es necesario pasar por cada id para obtener todos aquellos libros que ha jugado un usuario. En la función, la id del libro se guarda en la variable *numero* la cual se usara para hacer una búsqueda después en el JSON de libros para poder imprimir el título de este.

4.4.9 ¿Qué es QWERTY!?



Figura 4-23. Escena que explica la finalidad de la web

En la escena de la **Figura 4-23** se describe que es el proyecto QWERTY!. Se dice quien es el creador, el motivo de su creación y una breve descripción de qué se encontrarán los usuarios en el sitio web.

Dispone de un botón el cuál llevara al usuario a la escena de registro o autenticación en caso de que no esté autenticado, o a la escena de selección de modo si está autenticado.

4.5 Scripts comunes a todas las escenas

En esta sección se explican los scripts comunes a todas las escenas que conforman el sitio web.

4.5.1 Incremento y decremento del tamaño de letra

Esta funcionalidad se encuentra en el menú izquierdo de *Accesibilidad* de todas las escenas. Permite aumentar y disminuir el tamaño de la letra de la web, según se pulse un botón u otro.

4.5.2 Modo oscuro

Esta funcionalidad se encuentra en el menú izquierdo de *Accesibilidad* de todas las escenas. Si se pulsa sobre el cerrojo, se cambiará el modo de visualización de la escena, de modo normal (claro), a modo oscuro. Por ejemplo, la visualización de la escena de la *Figura 4-10* se muestra en su modo oscura en la *Figura 4-24*.

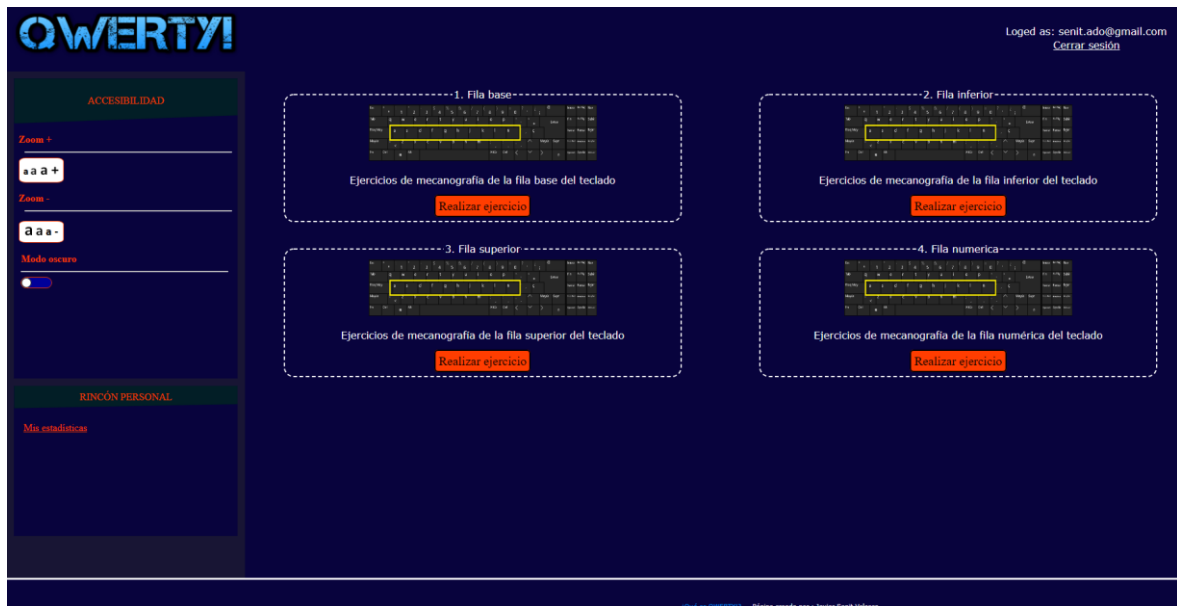


Figura 4-24. Ejemplo de modo oscuro en la escena de selección de temario

La parte complicada de esta función, es que el modo debe guardarse entre escena y escena, pues no tendría sentido cambiar el modo a oscuro, y al pasar a la siguiente escena, volviera a ser claro. Esto se realizó mediante la propiedad *sessionStorage* del objeto *Window* de javascript. Usando *sessionStorage.setItem('oscuro', 'blanco');* se consigue darle a la variable de sesión *oscuro*, el valor de *blanco*. Para obtener este valor solamente hace falta realizar un *data = sessionStorage.getItem('oscuro')*, el cual obtiene el valor de la variable de sesión oscuro.

4.5.3 Etiqueta *logged as* y cerrar sesión

Siempre y cuando estemos autenticados en la web, en la parte superior derecha de cada escena nos indicará que usuario esta autenticado, mostrándose su correo electrónico. Esto se consiguió a partir de una función que nos ofrece la librería Authentication de Firebase. La función que obtiene el usuario activo e imprime su valor es la que se muestra en la *Figura 4-25*.


```

firebase.auth().onAuthStateChanged(function(currentUser) {
  if (currentUser) {
    var user = currentUser

    document.getElementById("login").style.color="white";
    document.getElementById("login").innerHTML = "Logged as: " + user.email;
    document.getElementById("logout").innerHTML = "<a class='loginlogout' href='#'>Cerrar sesión</a>"
  }
});

```

Figura 4-25. Función que obtiene el usuario activo y lo escribe en la parte superior derecha

La función `onAuthStateChanged()` mira si existe algún usuario autenticado en ese momento. El usuario autenticado se guarda en la variable `currentUser`, y su propiedad `email` es la que se imprime en la parte superior derecha, aparte de imprimirse el link necesario para poder cerrar sesión.

En la **Figura 4-26** se muestra la función que cierra sesión, perteneciente también a la librería Authentication de Firebase:

```

firebase.auth().signOut().then(function() {
  window.location.href = "index.html";
}).catch(function(error) {
  alert("No se ha podido cerrar sesión correctamente");
});

```

Figura 4-26. Función que cierra la sesión de un usuario

Si se ha cerrado sesión correctamente se redirige al usuario a la escena principal.

5 Integración, pruebas y resultados

En este capítulo se especifican todas las pruebas desarrolladas para probar el correcto funcionamiento de la web. En primer lugar se habla de las pruebas de caja blanca y pruebas de caja negra. Posteriormente se han realizado pruebas de validación. Por último se hizo uso de una aplicación web, que analiza el código javascript y nos proporciona todos los warnings o defectos que tiene en lo que a código limpio se refiere.

5.1 Pruebas de caja negra

Estas pruebas se basan en comprobar el funcionamiento de la web sin tener en cuenta la lógica interna de la misma. Básicamente es comprobar si la finalidad con la que se desarrolló la web se cumple correctamente. Estas pruebas fueron realizadas de manera muy exhaustiva, y cada error era anotado para su posterior análisis. Surgieron un gran número de errores, pero todos ellos fueron subsanados de manera satisfactoria.

5.2 Pruebas de caja blanca

Estas pruebas comprueban la funcionalidad de cada uno de los módulos que conforman el proyecto completo. Estas dieron muchos más problemas, pues en realidad estas pruebas se iban realizando a medida que se finalizaba cada módulo, es decir, se acababa el módulo de registro y se le realizaban las pruebas de caja blanca. Los errores fueron solucionándose módulo tras módulo. En cuanto a funcionalidad todos fueron resueltos pero aún quedan warnings que no se han podido solucionar en alguna función.

5.3 Pruebas de validación

Estas pruebas se basan en comprobar que requisitos se han cumplido y cuáles no. Para este proyecto, se han conseguido cumplir con éxito aproximadamente el 87% de los requisitos, pues se han cumplido 26 de los 30 requisitos que se tienen (teniendo en cuenta funcionales y no funcionales). Los no cumplidos (explicados en el apartado *Resolución de requisitos*, del capítulo de *Diseño*) se tratarán en el capítulo de trabajo futuro, pues se arreglarán completamente.

5.4 Validación de código limpio

Para asegurar que el código javascript implementado fuese lo más limpio y ordenado posible se utilizó una herramienta online llamada JSHint [\[16\]](#). En esta herramienta simplemente pegas el código en un cuadro de texto que tiene, y ella misma te dice aquellos warnings en cuando a código limpio se refiere. En la **Figura 5-1** se muestra el ejemplo del análisis del código javascript del fichero *registro.js* :

```
CONFIGURE

Metrics
There are 16 functions in this file.
Function with the largest signature take 1 arguments, while the median is 0.
Largest function has 6 statements in it, while the median is 3.
The most complex function has a cyclomatic complexity value of 6 while the median is 1.

Eight warnings
  2  Missing semicolon.
  4  Missing semicolon.
264  'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
265  'let' is available in ES6 (use 'esversion: 6') or Mozilla JS extensions (use moz).
270  Expected a conditional expression and instead saw an assignment.
276  Missing semicolon.
279  Missing semicolon.
280  Missing semicolon.

Three undefined variables
  16  firebase
 124  firebase
 155  firebase
   21  curSize
   30  curSize

Nine unused variables
   2  errorUsuario
   3  oscuro
  19  zoomIn
  28  zoomOut
 134  registerUsername
 152  login
```

Figura 5-1. Resultados tras pasar la herramienta JSHint al código de registro

La herramienta web muestra un resumen de todo lo que considera que no es código limpio. Detecta warnings, variables no definidas y variables no usadas. Este análisis se pasó a todo el código del proyecto y se solucionaron todos los problemas existentes, dejando un código limpio y elegante.

6 Conclusiones y trabajo futuro

En este capítulo se explican las diversas conclusiones a las que se ha llegado durante el desarrollo completo del proyecto, así como que futuros cambios y nuevas funcionalidades se podrán implementar para mejor lo conseguido.

6.1 Conclusiones

Este proyecto comenzó con una idea bastante clara de que es lo que se quería realizar, un sitio web que permitiese enseñar a los usuarios como mecanografiar de manera correcta. Se pretendía crear una web que pudiese ser usada en colegios e institutos para la enseñanza y el aprendizaje.

A medida que se iba implementando esta funcionalidad, surgían nuevas ideas, como por ejemplo la idea de juego online, el cuál ayudaría a asentar todos los conocimientos obtenidos en el curso.

Se ha conseguido realizar una web donde se juntan ambos apartados, un curso online para que los usuarios aprendan a mecanografiar paso a paso, sin límite de tiempo y sin presión, y un juego donde poder aplicar todos los conocimientos adquiridos. Además, todo esto se ha conseguido mediante la implementación de código javascript nativo, por lo que todo aquel programador con nociones básicas de este lenguaje podrá entender cómo se han realizado todas y cada una de las funcionalidades, en parte también porque se ha conseguido que el código quede limpio y fácil de entender. Todo esto hace que la aplicación sea viable para usarse en entornos como la enseñanza y la educación como apoyo al resto del aprendizaje de los usuarios.

6.2 Trabajo futuro

Los cambios más inmediatos que se le realizarán a la aplicación, será la implementación de los requisitos iniciales que no fueron cumplidos satisfactoriamente, pues 4 de los 30 requisitos que se especificaron no fueron realizados.

Una nueva implementación que se realizará será el lanzamiento de una aplicación de escritorio, con la misma funcionalidad, de manera que no se necesite ningún navegador para poder realizar el curso ni para jugar online. Esta aplicación será realizada en Java o en Python, pues son lenguajes muy universales y en los cuáles es sencillo implementar código para la realización de una aplicación.

Se implementará también un buzón de sugerencias, donde los usuarios podrán escribir a los administradores de la web indicando posibles problemas o nuevas ideas para mejorar la web.

Por último, destacar que al ser una aplicación dirigida a muchos usuarios, la parte de interfaz y aspecto visual se irá modificando con el fin de mejorar la experiencia de todos los usuarios.

Referencias

- [1] Digishuffle (2019, Oct 30). 2018 Google Benchmarks: Bounce Rate & Avg. Session Duration. [Online]. Available: <https://www.digishuffle.com/blogs/bounce-rate-session-duration-benchmarks/>
- [2] Moodle [Online] Available: <https://moodle.org/?lang=es>
- [3] Statista. (2020, May 19). España, tercer país de la UE donde se realizan más cursos en línea. [Online] Available: <https://es.statista.com/grafico/21745/porcentaje-de-personas-que-han-realizado-un-curso-en-linea-en-los-ultimos-3-meses-en-paises-seleccionados/>
- [4] Javascript (2009, Sep 23). Douglas Crockford on Functional JavaScript. [Online] Available: https://web.archive.org/web/20090923001111/http://www.blinkx.com/video/douglas-crockford-on-functional-javascript/xscZz8XhfuNQ_aaVuyUB2A
- [5] Desarrolloweb (2016, Nov 29). Qué es una SPA. [Online]. Available: <https://desarrolloweb.com/articulos/que-es-una-spa.html>
- [6] ARTypist. ARTypist, el sitio idóneo para aprender mecanografía y mejorar tu velocidad.[Online] Available: <https://www.artypist.com/es/>
- [7] TypingClub. Learn Touch Typing for free! [Online] Available: <https://www.typingclub.com/>
- [8] Keyhero. Prueba de mecanografía. [Online] Available: <https://www.keyhero.com/prueba-mecanografia/>
- [9] Velocidactil. Juego online de mecanografía. [Online] Available: <https://www.velocidactil.es/>
- [10] Pmoinformatica (2018, May 30). ¿Qué es un requerimiento funcional? [Online]. Available: <http://www.pmoinformatica.com/2018/05/que-es-requerimiento-funcional.html>
- [11] Medium (2018, Apr 20). Requerimientos Funcionales y No Funcionales, ejemplos y tips. [Online]. Available: <https://medium.com/@requeridosblog/requerimientos-funcionales-y-no-funcionales-ejemplos-y-tips-aa31cb59b22a>
- [12] Rockcontent (2019, Apr 16). ¿Ya conoces Firebase? La herramienta de desarrollo y análisis de aplicaciones mobile. [Online]. Available: <https://rockcontent.com/es/blog/que-es-firebase/>
- [13] Kapitalsoluciones (2016, Jun 24). Psicología del color: significado y curiosidades de los colores. [Online] Available: <http://www.kapitalsoluciones.com/single-post/2016/06/24/Psicolog%C3%ADa-del-color-significado-y-curiosidades-de-los-colores>
- [14] Firebase . Estructura de tu base de datos. [Online]. Available: <https://firebase.google.com/docs/database/ios/structure-data?hl=es>
- [15] Firebase. Recupera Datos. [Online]. Available: <https://firebase.google.com/docs/database/admin/retrieve-data?hl=es-419>
- [16] JShint. [Online]. Available: <https://jshint.com/>

Glosario

JavaScript	Lenguaje de programación orientado al desarrollo web.
Firebase	Plataforma proporcionada por Google que ofrece herramientas para la gestión de bases de datos en la nube.
Bounce Rate	Porcentaje de rebote
JSHint	Herramienta tanto online como en aplicación que analiza código en busca de fallos en cuanto a la limpieza de éste.
HTML	Lenguaje de marcas de hipertexto.
CSS	Lenguaje de diseño gráfico

Anexos

A Manual de usuario

En este anexo se explicará al usuario el funcionamiento de la web QWERTY! con el fin de facilitar su experiencia durante su estancia en la web.

La escena que se muestra nada más a acceder a QWERTY! es la escena de autenticación y registro, tal como muestra la **Figura A-1**. Si el usuario no tiene ninguna cuenta asociada a la web, debe pulsar sobre el enlace Ingresa aquí, el cual mostrará el correspondiente formulario de registro. Cuando se rellene de manera correcta todos los campos especificados, se redirige al usuario al formulario de autenticación.

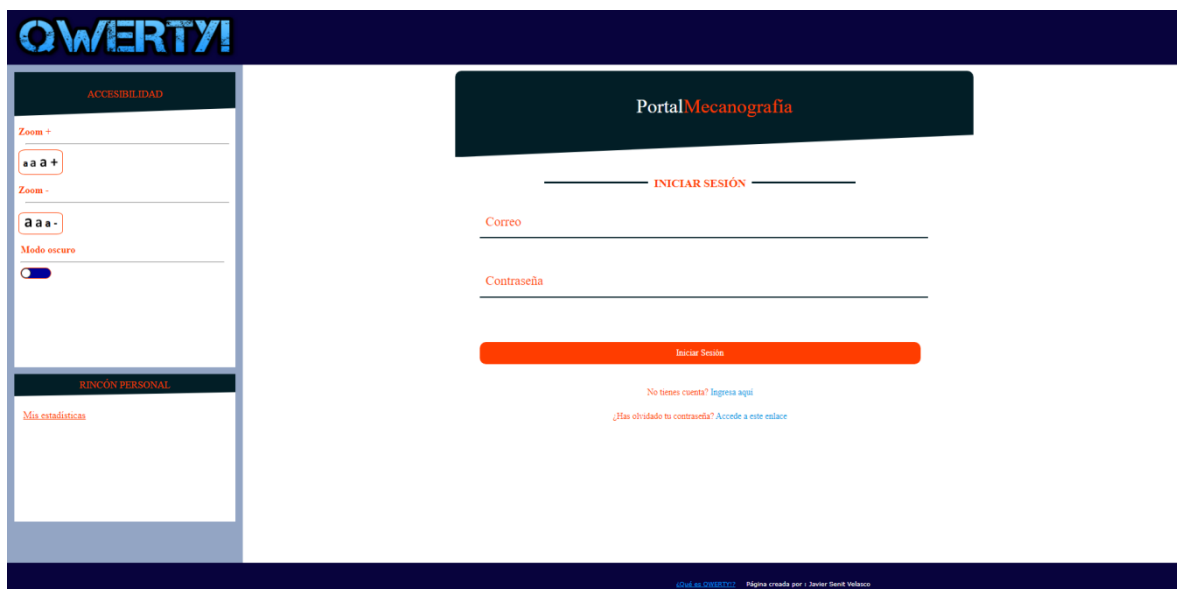
The screenshot shows the 'PortalMecanografia' login and registration interface. On the left is a sidebar with an 'ACCESIBILIDAD' (Accessibility) section containing 'Zoom +', 'Zoom -', and 'Modo oscuro' (Dark mode) toggle. Below this is the 'ESPACIO PERSONAL' (Personal Space) section with a link to 'Mis estadísticas'. The main content area features a dark header with the 'PortalMecanografia' logo. Below the header is a light blue box with the text 'INICIAR SESIÓN'. Underneath are input fields for 'Correo' (Email) and 'Contraseña' (Password). A large orange button labeled 'Iniciar Sesión' is positioned below the input fields. At the bottom of the form area, there are two links: 'No tienes cuenta? Ingresa aquí' (Don't have an account? Sign up here) and '¿Has olvidado tu contraseña? Accede a este enlace' (Forgot your password? Click this link). The footer of the page contains the text '© 2023 QWERTY! - Página creada por: Javier Sanz Velasco'.

Figura A-1. Escena de Autenticación y registro del usuario

Una vez ha introducido su correo electrónico y su contraseña, el usuario quedará autenticado en la web.

También se dispone del enlace que permite restablecer la contraseña en caso de haberse olvidado. Este enlace enviará al correo electrónico del usuario un vínculo que llevará al usuario a una página de Firebase donde se le solicitará un correo válido antes de poder establecer una nueva contraseña para el mismo.

Una vez dentro de la web, el usuario se encuentra en la escena de la **Figura A-2**, donde puede observar dos enlaces. Si lo que se desea es acceder al curso, se pulsará sobre dicho enlace, si por el contrario se desea entrar al modo de juego online, se pulsará sobre el segundo botón. Esto último no es recomendable para usuarios principiantes, pues se enfrentará a textos muy largos y a usuarios más avanzados y con mejor técnica.



Figura A-2. Escena de selección de modo

Si el usuario pulsa sobre *Curso online* se le redirige a la escena de la **Figura A-3**, correspondiente con la selección del temario para la realización de ejercicios.

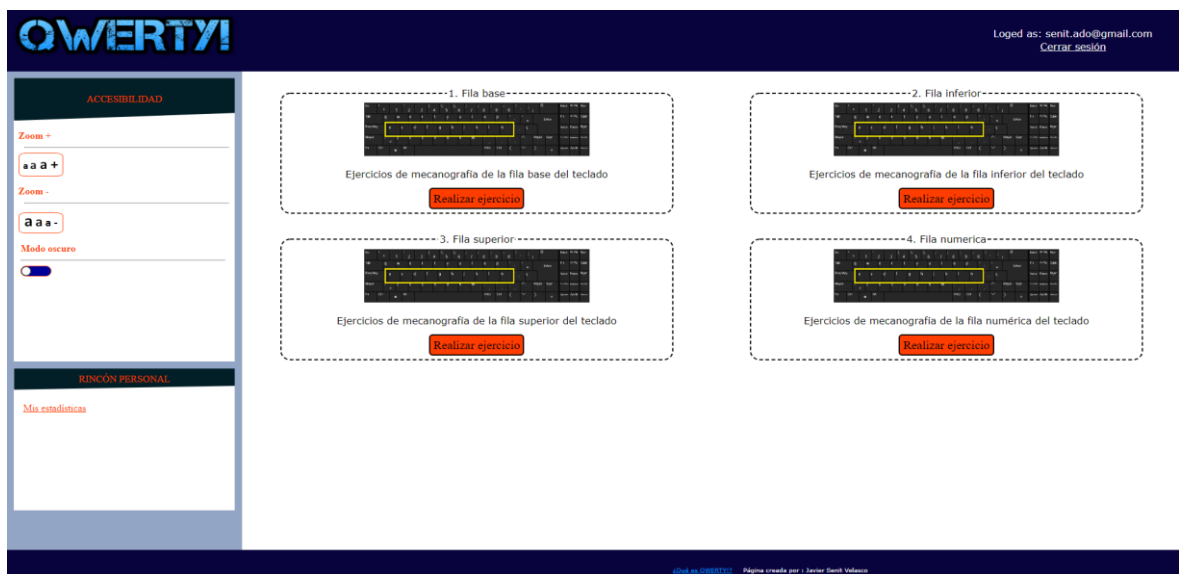


Figura A-3. Escena de selección de temario

En esta escena el usuario podrá escoger entre 4 temas diferentes. Podrá empezar a practicar con la fila central del teclado, con la superior, con la inferior o con la numérica. Cabe destacar que los ejercicios estarán basados en el lenguaje castellano, pues la mecanografía en otros idiomas es completamente diferente.

QWERTY!

Logged as: senit.ado@gmail.com
Cerrar sesión

ACCESIBILIDAD

Zoom +

aaa+

Zoom -

aaa-

Modo oscuro

☒

Función Personal

Min estadísticas

SELECCIONA UN EJERCICIO

Ejercicios básicos

Ejercicios intermedios

Ejercicios avanzados

Ejercicio básico 1

Ejercicio básico 2

Ejercicio básico 3

Ejercicio básico 4

Ejercicio básico 5

© 2023 QWERTY! | Página creada por: Javier Senit Vilasco

Una vez allí el usuario puede observar que existen tres tipos de ejercicios, básicos, intermedios y avanzados. Si se pone el cursor sobre cualquiera de ellos se despliegan todos los ejercicios de ese tipo.

Owerty!

Logged as: senit.ado@gmail.com
[Cerrar sesión](#)

ACCESIBILIDAD

Zoom +
aaa+


Zoom -
aaa-

Modo oscuro

RINCÓN PERSONAL

Mis estadísticas

Recuerde la posición inicial de las manos



ñe ñe meñ e ñf ñe jñj jñ jñ
jñ jñ

Escriba aquí el texto de arriba. El tiempo empezará a contar una vez teclee la primera letra.

¡Únete a OWERTY!

Página creada por : Javier Saez Vidales

Aquí el usuario debe escribir el texto indicado en la caja grande. Si se realiza sin errores y con una media de pulsaciones por segundo igual o superior a 5 se considerará como ejercicio aprobado. Si por el contrario, se cometen o errores o no se llegan a las

pulsaciones por segundo mínima, se considerará como ejercicio suspenso, y el usuario deberá volver a realizarlo. Un ejemplo de esto se muestra en la **Figura A-6**.

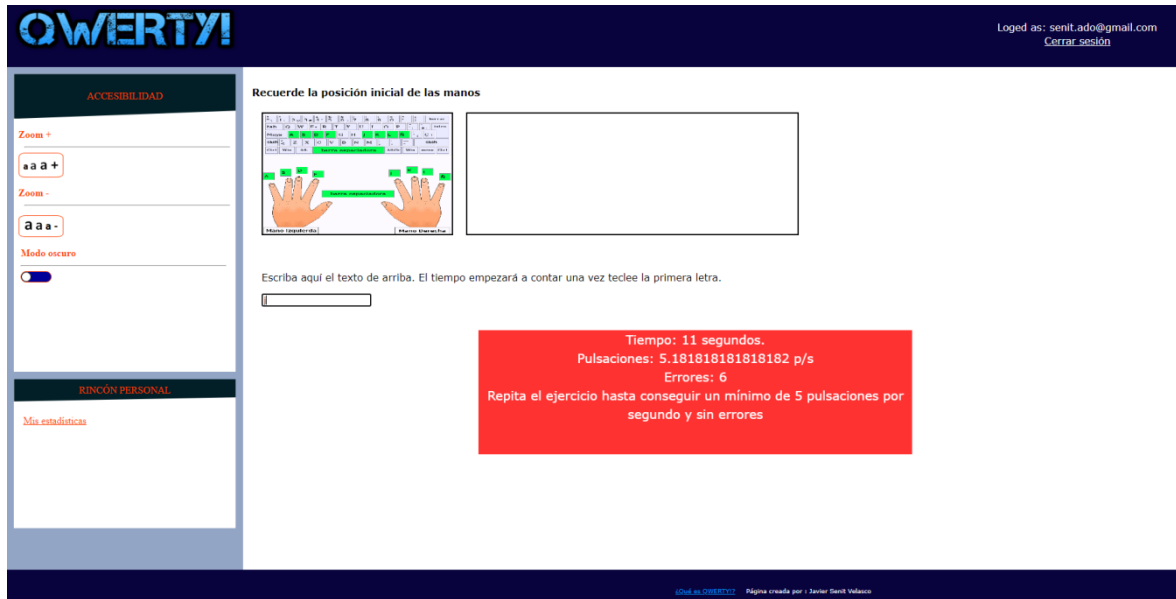


Figura A-6. Ejemplo de ejercicio no superado.

Si en la escena de selección de modo se pulsa sobre el juego online se lleva al usuario a una escena muy parecida a la anterior. La escena de la **Figura A-7** muestra como sería un texto largo del modo de juego online.



Figura A-7. Escena de texto largo del modo de juego online.

Cuando el usuario termine de escribir el texto, se le proporciona información de retroalimentación y sus datos se guardan para el ranking online.

Cuando el usuario pulse el enlace Ver ranking, se le redirige a la escena de la Figura A-8 donde podrá ver el ranking correspondiente al texto que acaba de escribir, pudiendo ver la posición en la que se encuentra.



Figura A-8. Escena de ranking online de un texto determinado.

Al usuario que esté activo se le marca en negrita sus resultados para que se encuentre más fácilmente en el listado.

Si el usuario pulsa sobre el enlace que se encuentra en el Rincón Personal, podrá acceder a todos sus resultados online. Esto se muestra en la Figura A-9. En esta escena se muestran todos los textos largos que ha escrito el usuario en el modo online, y puede acceder directamente al ranking de ellos pulsando sobre el título del texto.



Figura A-9. Escena de estadísticas generales de un usuario.

